目 录

第1章 CLI命令行介绍

本章对 CLI **命令行接口进行详细的描述，主要包括以下内容：**

- **访问交换机的**CLI
- CLI**模式介绍**
- **命令语法介绍**
- **命令行快捷键**
- **历史命令**

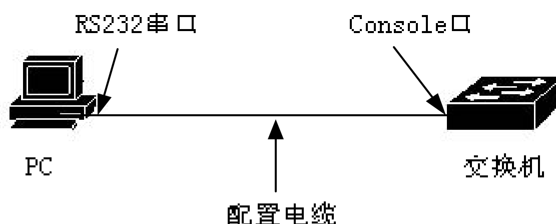
1.1 访问交换机的 CLI

交换机的 CLI 命令行接口提供了用户管理交换机的界面。用户可以通过 Console 口和 Telnet 两种终端来访问交换机的 CLI 命令行接口，下面分别介绍。

1.1.1 用户通过 Console 口访问 CLI

操作步骤如下：

第一步：通过配置电缆把 PC 的串口与交换机的 Console 口连接，如下图：



第二步：启动PC 机上的终端仿真程序（如 Windows 的超级终端等），配置终端仿真程序的通信参数。终端的通信参数配置如下：

波特率：38400

数据位：8

奇偶校验： 无

停止位：1

数据流控制： 无

超级终端的通信参数配置如下图：



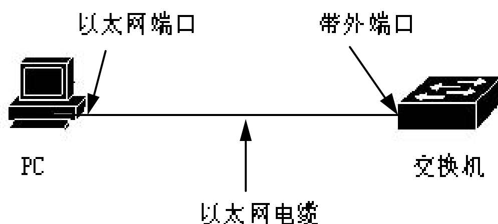
第三步：启动交换机，交换机启动完成后会在终端上显示 CLI 提示符（缺省为 Switch>），用户可以在此提示符下输入命令，这样用户就可以访问交换机的 CLI 了。

1.1.2 用户通过 TELNET 访问 CLI

用户可以通过 iSpirit 12800 交换机的管理模块的带外端口或接口模块的带内端口访问交换机。

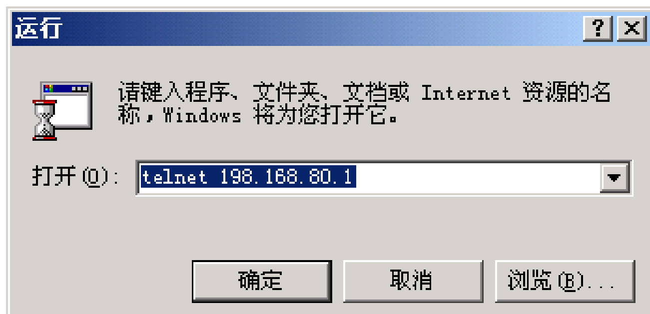
交换机的带外端口的 IP 地址缺省为 198.168.80.1，通过带外端口访问交换机的操作步骤如下：

第一步：通过以太网电缆（必须为交叉线）连接 PC 机的以太网端口和交换机的带外端口。如下图：



第二步：设置 PC 机的以太网端口的 IP 地址，该 IP 地址必须在 198.168.80.0/24 段内（如 IP 地址 198.168.80.100）。通过 ping 198.168.80.1 来判断 PC 机与交换机的连通性。

第三步：如果 PC 机与交换机是连通的，则 Telnet 198.168.80.1 进入 Telnet 终端界面。如下图：



第四步：如果系统没有设置密码，Telnet 界面直接进入 CLI，出现 CLI 提示符（缺省为 Switch>）；如果系统设置了密码，在 Telnet 界面上需要输入密码后才能进入 CLI。

用户通过带内端口访问交换机的方法与通过带外端口访问交换机的类似，但有两点要特别注意：

- **带内端口的 IP 地址是建立在 VLAN 三层接口之上的，在访问交换机之前，必须设置某个 VLAN 接口的 IP 地址，VLAN1 的缺省 IP 地址是 192.168.0.1，可以直接使用。VLAN 接口的 IP 地址可以通过 Console 口或带外端口 Telnet 界面进行配置。**
- **用户通过带内端口访问交换机，可以直接通过以太网电缆连接 PC 和带内端口，也可以通过一个网络进行连接，只需要 PC 与交换机的某个 VLAN 之间能够互通就行了。**

1.2 CLI 模式介绍

1.2.1 CLI 模式的作用

CLI 模式的作用主要有如下两点：

- **方便对用户的分级，防止未授权的用户非法使用 CLI。**

用户可分为三个级别，也就是三类：普通用户，特权用户和配置用户。

普通用户只能查看交换机的一些运行状态，只能使用显示命令。

特权用户除了能够查看交换机的运行状态以外，还可以对交换机进行必要的维护，如对交换机进行调试，升级和存储等。

配置用户除了特权用户能做的事情以外,还可以对交换机进行配置,改变交换机的行为。

- 方便用户对交换机进行配置

交换机存在很多的配置,如果把所有的配置放在一个模式中,用户使用起来非常不方便。为此，在 CLI 上建立多个模式，把相近的命令放在一个模式中，便于用户的理解和使用。如把与 VLAN 相关的命令放在 VLAN 配置模式中，把与接口相关的命令放在接口配置模式中。

1.2.2 CLI 模式的标识

CLI 提示符是 CLI 模式的标识，用户在使用 CLI 时，通过看 CLI 提示符就知道目前所处的 CLI 模式。

CLI 提示符由两部分组成，一部分标识主机，另一部分标识模式。

CLI 提示符中的主机部分使用的是系统的主机名，系统的主机名是可配置的，缺省为 Switch，所以 CLI 提示符缺省是以 Switch 开头，后面提到的 CLI 描述符一般情况都使用缺省的主机名。

CLI 提示符中的模式部分是不可配置的，每种模式都有自己对应的模式字符串，有些模式字符串是固定不变的，而有些模式字符串是可变的。如 VLAN 配置模式的模式字符串是固定的，接口配置模式的模式字符串是可变的。

例如：

CLI 提示符 Switch#标识特权模式，Switch 标识主机，而#标识模式。

CLI提示符Switch(config-ge1/1)#标识接口配置模式,并且配置的是 ge1/1 端口,Switch 标识主机，而(config-ge1/1)#标识模式。

CLI 提示符 Switch(config-vlan2)#标识接口配置模式,并且配置的是 vlan2 接口,Switch 标识主机，而(config-vlan2)#标识模式。

1.2.3 CLI 模式的分类

CLI 模式分为普通模式，特权模式，全局配置模式和配置子模式四大类，而配置子模式由很多个 CLI 模式组成。

普通用户只能访问普通模式，特权用户可以访问普通模式和特权模式，配置用户可以访问所有的 CLI 模式。

Console 和 Telnet 终端首先进入的是普通模式，在普通模式下输入 enable 命令并且成功验证密码后进入特权模式。在特权模式下输入 configure terminal，系统判断此用户是否是配置用户，如果是配置用户则进入全局配置模式。在全局配置模式下输入相关的命令可以进入各配置子模式。

下表列出了交换机的主要的 CLI 模式：

模式	描述	提示符	进入模式的命令	退出模式的命令
普通模式	提供了显示命令查看交换机的状态信息。	Switch>	终端首先进入的模式。	在 Console 终端上没有退出模式的命令，在 Telnet 终端上使用 exit 或 quit 命令退出 Telnet 终端。
特权模式	除了提供显示命令查看交换机的状态信息外，还提供了调试，版本升级和配置维护等命令。	Switch#	在普通模式下输入 enable 命令。	使用 disable 命令退回到普通模式。在 Console 终端上使用 exit 或 quit 命令退到普通模式，在 Telnet 终端上使用 exit 或 quit 命令退出 Telnet 终端。
全局配置模式	提供了不能在配置子模式内实现的通用命令，如配置静态路由命令。	Switch(config)#	在特权模式下输入 configure terminal 命令。	使用 exit，quit 或 end 命令退出到特权模式。
接口配置模式	提供了配置端口和 VLAN 接口的命令。端口又可分为百兆，千兆和万兆端口。	百兆端口：Switch(config-fe1/1)# 千兆端口：	在全局配置模式下输入 interface <if-name>	使用 exit 或 quit 命令退出到全局配置模式，使用 end

		Switch(config-ge2/1)# **万兆端口：** Switch(config-xe3/1)# VLAN **接口：** Switch(config-vlan1)#	**命令。**	**命令退出到特权模式。**
VLAN **配置模式**	**提供了配置** VLAN **的命令。例如创建和删除** VLAN **的命令。**	Switch(config-vlan)#	**在全局配置模式下输入** vlan database **命令。**	**使用** exit **或** quit **命令退出到全局配置模式，使用** end **命令退出到特权模式。**
MSTP **配置模式**	**提供了配置** MSTP **的命令。例如创建和删除** MSTP **实例的命令。**	Switch(config-mst)#	**在全局配置模式下输入** spanning-tree mst configuration **命令。**	**使用** exit **或** quit **命令退出到全局配置模式，使用** end **命令退出到特权模式。**
RIP **配置模式**	**提供了配置** RIP **协议的命令，例如指定启动** RIP **的** IP **网段的命令。**	Switch(config-rip)#	**在全局配置模式下输入** router rip **命令。**	**使用** exit **或** quit **命令退出到全局配置模式，使用** end **命令退出到特权模式。**
OSPF **配置模式**	**提供了配置** OSPF **协议的命令，例如指定启动** OSPF **的** IP **网段的命令。**	OSPF **进程** 0 **的提示符：** Switch(config-ospf)# OSPF **进程** 1 **的提示符：** Switch(config-ospf-1)#	**在全局配置模式下输入** router ospf [ospf-id] **命令。**	**使用** exit **或** quit **命令退出到全局配置模式，使用** end **命令退出到特权模式。**
VRRP **配**	**提供了配置** VRRP **协议的命令，例如指定** VRRP **的**	Switch(config-vrrp)#	**在全局配置模式下输入**	**使用** exit **或** quit **命令退出到全局配**

置模式	虚拟 IP 地址的命令。		router vrrp <if-name> <vrid> 命令。	置模式，使用 end 命令退出到特权模式。
终端配置模式	提供了配置 Console 和 Telnet 终端的命令，如配置终端的超时时间的命令。	Switch(config-line)#	在全局配置模式下输入 line vty 命令。	使用 exit 或 quit 命令退出到全局配置模式，使用 end 命令退出到特权模式。
模块配置模式	提供了配置接口模块的命令，如创建和删除模块的命令等。	Switch(config-module)#	在全局配置模式下输入 module-management 命令。	使用 exit 或 quit 命令退出到全局配置模式，使用 end 命令退出到特权模式。
密钥链配置模式	提供了配置密钥链的命令，如创建和删除密钥链的命令等。	Switch(config-keychain)#	在全局配置模式下输入 key chain <name> 命令。	使用 exit 或 quit 命令退出到全局配置模式，使用 end 命令退出到特权模式。

1.3 命令语法介绍

1.3.1 命令组成

CLI 命令由关键字和参数两部分组成，第一个词必须是关键字，后面的词可以是关键字也可以是参数，关键字和参数可以交替出现。一个命令必须有关键字，但可以没有参数。例如命令 write 就只有一个关键字而没有参数；命令 show version 有两个关键字而没有参数；命令 vlan <vlan-id> 有一个关键字并且有一个参数；命令 instance <instance-id> vlan <vlan-id>有两个关键字和两个参数并且关键字和参数是交替出现的。

1.3.2 参数类型

CLI 命令的参数分为两种： 必选参数和可选参数。在输入命令时必选参数必须输入，而可选参数可以输入也可以不输入。如命令 vlan <vlan-id>中的参数是必选参数，在输入命令时此参数必须输入；而命令 show interface [if-name] 中的参数是可选参数，在输入命令时此参数可输入，也可不输入。

1.3.3 命令语法规则

在用文本描述命令时必须满足以下规则：

1）关键字直接用单词表示。

如命令 show version。

2）参数必须用< >括起来。

如命令 vlan <vlan-id>

3）如果是一个可选参数，参数必须用[ ]括起来。

如命令 show vlan [<vlan-id>]

对于这种情况，参数的< >可以省略，改成：

命令 show vlan [vlan-id]

也就是参数 vlan-id 可以输入，也可以不输入。

如果是一个必选参数，参数不能有[ ]。

4）如果有多个关键字或参数中必须选择一个，用{ }把多个关键字或参数括起来，多个关键字或参数之间用 | 隔开， | 前后都需要一个空格。

如多个关键字必选的命令：

spanning-tree mst link-type {point-to-point | shared}

在 point-to-point 和 shared 之间必须选择一个。

多个参数必选的命令：

no arp {<ip-address> | <ip-prefix>}

关键字和参数混杂必选的命令：

vrrp authentication {none | simple-password <password>}

5）如果多个关键字或参数中可选一个，用[ ]把多个关键字或参数括起来，多个关键字或参数之间用 | 隔开， | 前后都需要一个空格。

命令如下：

debug rip packet [recv | send]

关键字 recv和 send 可以选择一个，也可以不选。

show ip route [<ip-address> | <ip-prefix>]

show interface [<if-name> | switchport]

6）如果有一个关键字或参数或一组关键字或参数可以重复选择输入，在这个（组）关键字或参数后加符号"*"。

例如 ping 命令：

ping <ip-address> [-n <count> | -l <size> | -r <count> | -s <count> | -j <count> <ip-address>* | -k <count> <ip-address>* | -w <timeout>]*

-j <count> <ip-address>* --- 可以重复输入多个 IP 地址

-k <count> <ip-address>* --- 可以重复输入多个 IP 地址

整个选项也可以重复输入。

6）参数用一个或多个单词的描述符表示，如果是多个单词，用符号"-"隔开每个单词，每个单词都是小写。

正确的参数表示法：<vlan-id>，<if-name>，<router-id>， <count>等。

错误的参数表示法：<1-255>，<A.B.C.D>，<WORD>，<IFNAME>等。

1.3.4 命令缩写

用户在 CLI 界面上输入命令时，命令的关键字可以缩写。CLI 支持命令的前缀匹配功能，只要输入的词与关键字前缀唯一匹配，CLI 就把输入的词解析成匹配的关键字。这样用户在使用 CLI 时非常方便，用户可以键入很少的字符完成一个命令，例如 show version 命令可以只键入 sh ver。

1.3.5 语法帮助

CLI 命令行接口中设置有语法帮助,支持每一级命令和参数的帮助功能,分别描述如下：

1）在某个 CLI 模式下直接输入？键，在终端上会列出该模式下的所有命令的第一个关键字及其描述。例如 Switch(config)#?。

2）输入一个命令中的前面的部分，然后输入空格后再输入？键，在终端上会列出下一级的所有关键字或参数及其描述。例如 Switch#show ?。

3）输入一个不完整的关键字后直接输入？键，在终端上会列出与此输入前缀匹配的所有关键字及其描述。例如 Switch#show ver?。

4）输入一个命令中的前面的部分，然后输入空格后再输入 Tab 键，在终端上会列出下一级的所有关键字，下一级如果是参数，则不会列出来。

5）输入一个不完整的关键字后直接输入 Tab 键，如果只有一个关键字与此输入前缀匹配，则直接补齐，如果有多个关键字与此输入前缀匹配，则在终端上列出所有匹配的关键字。

1.3.6 命令行错误信息

用户输入的命令如果没有通过语法检查，会在终端上显示错误信息，常见的错误信息如下表。

错误信息	错误原因
Invalid input 或 Unrecognized command	没有找到匹配的关键字。 参数输入不对。 输入的关键字或参数太多。
Incomplete command	命令输入不完整，还有关键字或参数没有输入。
Ambiguous command	关键字输入不完整，有多个关键字与输入前缀匹配。

1.4 命令行快捷键

1.4.1 行编辑快捷键

CLI 命令行接口支持行编辑快捷键功能，行编辑快捷键可以方便 CLI 命令的输入和编辑。用户在输入或编辑命令时，可以使用行编辑快捷键加速命令的输入。下表列出所有的行编辑快捷键及实现的功能：

快捷键	功能
Ctrl+p 或?键	上一条命令
Ctrl+n 或?键	下一条命令
Ctrl+u	删除整行
Ctrl+a	光标回到行首
Ctrl+f 或? 键	光标向右移动一格
Ctrl+b 或? 键	光标向左移动一格
Ctrl+d	删除光标所在的字符
Ctrl+h	删除光标前一个字符
Ctrl+k	删除光标处及光标后的所有字符
Ctrl+w	删除光标前的所有字符
Ctrl+e	光标移到行尾
Ctrl+c	中断，不执行命令行。如果 CLI 处在全局配置模式或者是配置子模式，CLI 退到特权模式；如果 CLI 处在普通模式或特权模式，CLI 模式保持不变，但 CLI 另起新行。
Ctrl+z	与 Ctrl+c 功能相同。
Tab	输入不完整的关键字后使用此键，如果有一个关键字与输入的前缀匹配，则补齐此关键字；如果有多个关键字与输入的前缀匹配，则列出所有匹配的关键字；如果没有关键字匹配，则此键无效。

注意：有些 Console 终端上?、?、?、? 键不可用。

1.4.2 显示命令快捷键

对于以 show 关键字开头的命令都是显示命令，有些显示命令由于显示的内容很多，在一屏中无法显示完，终端提供了分屏显示的功能。在显示一屏后终端等待用户输入来决定后面的处理。下表列出了显示命令快捷键及其功能。

快捷键	功能
空格 Space	显示下一屏
回车 Enter	显示下一行
Ctrl+c	中断命令的执行，退出到 CLI 模式下。
其它键	与 Ctrl+c 功能相同。

1.5 历史命令

CLI 命令行接口支持命令的历史记录功能，能记住用户最近使用的 20 个历史命令，把用户最近键入的命令保存起来。您可以用 show history 来显示已经输入过的命令，您也可以用 Ctrl+p,Ctrl+n 或?、?键来选择历史命令。历史命令功能可以方便用户输入命令。

第2章 系统管理配置

用户在学习交换机的相关功能配置之前,需要先掌握交换机的系统管理和维护方面的一些基本配置，本章就描述这些系统管理和维护的基本配置，主要包括以下内容：

- 系统安全配置
- 系统维护和调试
- 配置文件管理
- 软件版本升级

2.1 系统安全配置

为了防止非法用户入侵交换机，系统提供了几种系统管理安全方面的措施，主要包括：

- 用户密码控制
- enable密码控制
- Telnet服务控制
- 源IP地址控制

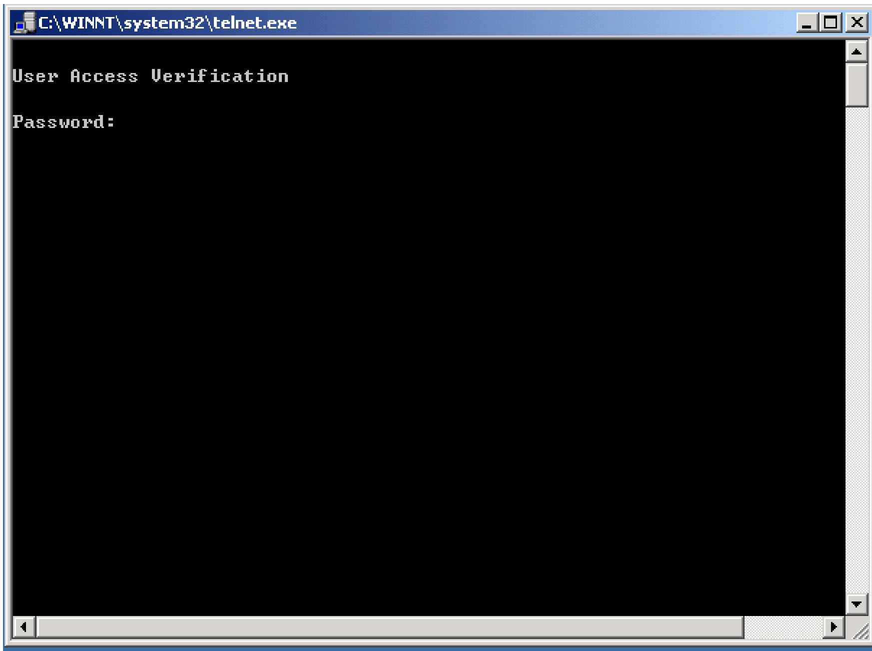
2.1.1 用户密码控制

系统目前只支持一个用户admin，还不支持多用户管理功能，admin用户是一个配置用户，具有最高的权限，可以访问所有的CL模式。

admin用户只用于控制Telnet终端，不控制Console终端。在使用Console终端访问交换机时不需要验证用户密码，用户直接可以访问CLI。而通过Telnet终端访问交换机时需要验证用户密码，只有密码验证成功后才能访问CLI。在通过Telnet终端访问交换机时不需要输入用户名，因为系统目前只有一个用户名，默认为admin。

admin用户的用户密码缺省为空，在这种情况下，用户访问Telnet终端时不需要进行用户密码验证。

下图为用户登陆Telnet终端的界面，在此界面下输入用户密码。



用户密码的相关命令如下表：

命令	描述	CLI模式
password <key>	设置admin用户的密码。	全局配置模式
no password	清除admin用户的密码，用户密码为空。	全局配置模式
show running-config	查看系统当前配置，可以查看到用户密码的配置。	特权模式

注意：为了系统的安全，管理员需要设置系统的用户密码。

2.1.2 enable 密码控制

enable密码用于控制普通模式到特权模式的切换，在 enable密码验证前，用户只能查看交换机的信息，而enable密码验证后，用户就有可能对交换机进行配置和维护。

enable密码不依附于用户，任何用户登陆到Console终端或Telnet终端，如果要进入特权模式都必须验证enable密码，如果验证不成功，则只能停留在普通模式下。

在普通模式下输入enable命令，终端会提示用户输入密码，此时用户可输入enable密码，如果密码验证成功，终端进入特权模式，否则，停留在普通模式，对于普通用户来说不管密

码是否验证成功都不能进入特权模式。

enable密码缺省为空，这种情况下在普通模式下输入enable命令后终端不提示输入密码直接进入特权模式。

enable密码的相关命令如下表：

命令	描述	CLI模式
enable password <key>	设置系统的enable密码。	全局配置模式
no enable password	清除系统的enable密码，enable密码为空。	全局配置模式
show running-config	查看系统当前配置，可以查看到enable密码的配置。	特权模式
enable	交互式命令，验证系统的enable密码，验证成功后，终端进入特权模式。	普通模式

注意：为了系统的安全，管理员需要设置系统的enable密码。

2.1.3 Telnet服务控制

在有些情况下，管理员不需要远程管理交换机，只需要在本地通过Console终端来管理交换机就行了，此时为了提高系统的安全性，防止非法用户远程登陆Telnet终端，管理员可以关闭Telnet服务。Telnet服务缺省是打开的。

Telnet服务控制的相关命令如下表：

命令	描述	CLI模式
enable telnet	打开Telnet服务。	终端配置模式
disable telnet	关闭Telnet服务。	终端配置模式
show running-config	查看系统当前配置，可以查看到Telnet服务控制的配置。	特权模式

2.1.4 源 IP 地址控制

在需要远程管理交换机的情况下，为了提高系统的安全性，可以通过源IP地址控制的方法来实现系统的安全管理。如果允许某个IP地址访问交换机，则使用该IP 地址的PC可以登陆Telnet终端，如果禁止某个IP 地址访问交换机，则使用该IP 地址的P C无法登陆Telnet 终端。

源IP地址控制要使用到ACL资源库，在实施之前，需要先配置标准ACL规则组，在规则组中定义一条或多条源IP地址的过滤规则。在配置源IP 地址控制时，直接使用ACL组名。

缺省情况下系统没有实施源IP 地址控制，任何与交换机能够连通的PC都能登陆到Telnet 终端。当然，在Telnet 服务是关闭的情况下，源IP 地址控制没有任何意义。

源IP 地址控制的相关命令如下表：

命令	描述	CLI模式
access-class <acl-name>	指定一个ACL组，打开源IP 地址控制，如果指定的ACL组不存在或不是标准ACL组，则不对源IP 地址进行控制。	终端配置模式
no access-class	关闭源IP地址控制。	终端配置模式
show running-config	查看系统当前配置，可以查看到源 IP地址控制的配置。	特权模式

2.2 系统维护和调试

基本的系统维护和调试功能主要包括以下内容：

- 配置系统的主机名
- 配置系统的时钟
- 配置终端超时属性
- 系统复位
- 查看系统信息

- 网络连通性调试
- Traceroute调试

2.2.1 配置系统的主机名

系统的主机名用于标识交换机，方便用户区分不同的交换机，同时系统的主机名还是终端的CLI提示符的一部分。系统的主机名缺省是Switch。

系统的主机名的相关命令如下表：

命令	描述	CLI模式
hostname <name>	设置系统的主机名。	全局配置模式
no hostname	清除系统的主机名,即主机名回到缺省值Switch。	全局配置模式
show running-config	查看系统当前配置,可以查看到系统的主机名的配置。	特权模式

2.2.2 配置系统的时钟

交换机提供了实时时钟的功能，通过命令可以设置当前时钟，也可以查看当前时钟。系统的时钟由内部供电，保证系统断电时实时时钟的持续运行，系统启动后不需要重新设置时钟。

交换机在出厂时已经设置好时钟，用户不需要再进行设置，如果用户发现时间不准时，用户可以重新设置时钟。

系统时钟的相关命令如下表：

命令	描述	CLI模式
set date-time <year> <month> <day> <hour> <minute> <second>	设置系统的当前时钟，需要输入年、月、日、小时、分和秒参数。	特权模式
show date-time	显示系统的当前时钟。	普通模式，特权模式

2.2.3 配置终端超时属性

为了终端的安全性，当终端没有键输入的情况下，超过一定的时间，终端会做退出处理。Console终端和Telnet终端的退出处理不一样，对于Console终端，当终端超时时，CLI模式退到普通模式，对于Telnet终端，当终端超时时，Telnet连接中断，Telnet终端退出。
终端超时时间缺省为10分钟，用户也可以设置终端永远不超时。
终端超时的相关命令如下表：

命令	描述	CLI模式
exec-timeout <minutes> [seconds]	设置终端超时时间，如果参数都为0时，表示终端永远不超时。	终端配置模式
no exec-timeout	设置终端超时时间回到缺省情况，即10分钟。	终端配置模式
show running-config	查看系统当前配置，可以查看到终端超时的配置。	特权模式

2.2.4 系统复位

系统提供了以下几种复位方法：

- 复位管理模块自身
- 复位某个接口模块
- 复位整个交换机系统

系统复位的相关命令如下表：

命令	描述	CL模式
reset	复位管理模块自身，不影响接口模块。	特权模式
reset module <module-id>	复位某个接口模块，对于iSpirit 12804交换机，参数是1到4，对于iSpirit 12810交换机，参数是1到8。	特权模式

reset system	复位整个交换机系统，包括所有在位的管理模块和接口模块。	特权模式

2.2.5 查看系统信息

系统提供了丰富的显示命令来查看系统的运行状态和系统的信息，这里只列出几个常用的系统维护的显示命令，如下表：

命令	描述	CL模式
show version show bdver	显示系统的版本号和执行文件编译连接的时间。 显示所有的管理模块和接口模块的版本号。	普通模式，特权模式 普通模式，特权模式
show system	显示系统的基本信息，包括系统启动后运行了多长时间。	普通模式，特权模式
show history	显示在CLI命令行上最近输入的命令列表。	普通模式，特权模式

2.2.6 网络连通性调试

为了调试交换机与网络中的另一设备的连通性，需要在交换机上实现ping命令，在交换机上ping对方的IP地址，如果交换机收到对方来的ping应答，说明两端是连通的，否则表明两端不能进行通信。

交换机不仅实现了ping命令，还在ping命令上支持很多选项，用户通过使用这些选项进行更加精确和复杂的调试。

ping命令如下表：

命令	描述	CLI模式
ping <ip-address> [-n <count> | -l <size> | -r <count> | -s <count> | -j <count> <ip-address>* | -k	在使用时可以不带任何选项，也可以带一个或多个选项。如果不带任何选项，就	特权模式

<count> <ip-address>* | -w <timeout>]*	是最简单的ping命令。命令在执行时可键入Ctrl+c中断命令的执行。	

2.2.7 Traceroute 调试

为了调试交换机与网络中的另一设备在通信时经过了哪些中间设备时，需要在交换机上实现trace-route命令。在交换机上使用trace-route命令时，指定对方的IP地址，命令执行过程中会把中间经过的路径全部显示出来。

交换机不仅实现了trace-route命令，还在trace-route命令上支持很多选项，用户通过使用这些选项进行更加精确和复杂的调试。

trace-route命令如下表：

命令	描述	CLI模式
trace-route <ip-address> [-h <maximum-hops> | -j <count> <ip-address>* | -w <timeout>]*	在使用时可以不带任何选项，也可以带一个或多个选项。如果不带任何选项，就是最简单的trace-route命令。命令在执行时可键入Ctrl+c中断命令的执行。	特权模式

2.3 配置文件管理

配置分为当前配置和初始配置两种。当前配置指的是系统运行时的配置，存在系统的内存中，而初始配置是系统启动时用到的配置，存在系统的FLASH中，也就是配置文件。当用户执行相关命令时修改的是系统的当前配置，只有执行了保存命令后才把当前配置写入到初始配置中，用于系统的下一次启动。当系统启动后用户在没有做任何配置的情况下，系统的当前配置信息与初始配置信息相同。

当前配置和初始配置采用相同的格式，都是命令行文本的格式，非常直观，便于用户阅读。配置文件的格式具有如下几个特点：

- 配置文件是文本文件。
- 保存的都是命令。
- 只保存非缺省的配置，对于缺省的配置不保存。
- 命令是按CLI模式来组织的，把同一个CLI模式下的命令组织在一起，形成一段，段与段之间以“!”隔开。对于全局配置模式内的命令，把同一功能或功能相近的命令组织成一段，以“!”隔开。
- 对于配置子模式内的命令，在命令前有一个空格，而对于全局配置模式内的命令，命令前不需要有空格。
- 以“end”作为配置的结束。

配置文件管理主要包括以下内容：

- 查看配置信息
- 保存配置
- 删除配置文件
- 配置文件上下载

2.3.1 查看配置信息

查看配置信息包括查看系统的当前配置和初始配置。初始配置实际上就是在FLASH中的配置文件，当FLASH中不存在配置文件时，系统启动时使用的是缺省配置，此时如果查看系统的初始配置，系统会提示配置文件不存在。

查看配置信息的命令如下表：

命令	描述	CLI模式
show running-config	查看系统的当前配置。	特权模式
show startup-config	查看系统的初始配置。	特权模式

2.3.2 保存配置

当用户修改了系统的当前配置，这些配置需要保存到配置文件中去，这样下一次启动后这些配置还依然存在，否则，重启后这些配置信息就丢失了。保存配置就是把当前配置保存到初始配置中。

保存配置的命令如下表：

命令	描述	CLI模式
write	保存当前的配置。	特权模式

注意：用户在对交换机做了配置后需要使用此命令保存配置，否则系统重启后配置会丢失。

2.3.3 删除配置文件

当用户希望系统的初始配置回到缺省配置时可以删除配置文件，删除配置文件后对当前配置没有影响，如果希望系统的当前配置回到缺省配置时，需要重启交换机。用户在做删除配置文件时一定要谨慎，否则配置会丢失。

删除配置文件的命令如下表，

命令	描述	CLI模式
delete startup-config	删除系统的配置文件。	特权模式

2.3.4 配置文件上下载

为了配置文件的安全性，用户可以使用命令把配置文件上载到PC机上做备份，当系统的配置异常丢失或做了修改后希望回到原来的配置时，可以从PC机上把原来的配置文件下载到交换机上，下载配置文件后对系统的当前配置没有影响，必须重启交换机后配置就能生效。

配置文件上下载的命令如下：

命令	描述	CLI模式
upload configure <ip-address> <file-name>	把配置文件上载到PC机上，第一个参数是PC机的IP地址，第二个参数是配置文件在PC机上存储的文件名。	特权模式

download configure <ip-address> <file-name>	把配置文件下载到PC机上，第一个参数是PC机的IP地址，第二个参数是配置文件在PC机上存储的文件名。	特权模式

配置文件上下载使用到TFTP 协议，在交换机上运行TFTP客户端软件，在PC机上运行TFTP服务端软件。配置文件上下载的操作步骤如下：

第一步： 搭建网络环境

第二步： 在PC机上启动TFTP服务端软件，设置配置文件所存放的目录。

第三步： 在交换机上保存配置。

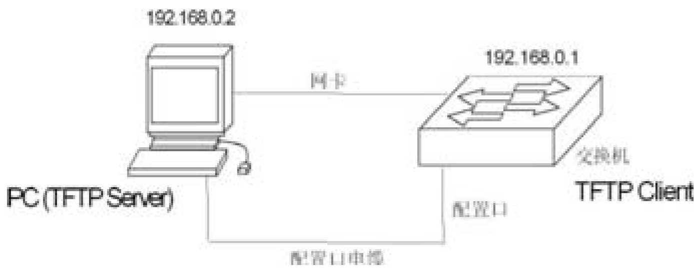
第四步： 在交换机上执行配置文件上载命令把配置文件备份到PC机上。

第五步：当交换机需要PC机上的配置文件时，在交换机上执行配置文件下载命令把PC机上的配置文件下载到交换机上。

第六步：要使配置生效，必须重启交换机。

示例： 一台已经配置好了VLAN和接口地址的交换机，需要进行配置文件上下载操作。

第一步： 搭建如下所示网络环境。



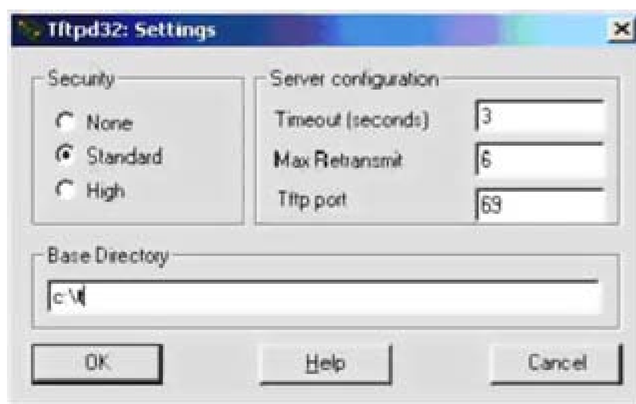
将交换机的配置口通过电缆外接一台配置终端，并通过网线与一台PC 相连。在PC 安装TFTP Server， 配置PC 的以太网口IP 地址，这里假定PC 的IP 地址为192. 168.0.2 。然后，配置交换机的IP 地址，这里假定交换机的IP 地址为192.168. 0 . 1 ，保证PC机与交换机之间的连通性。

第二步：启动TFTP Server，配置TFTP Server参数。

运行TFTP Server，窗口界面如下图：



然后，设置备份配置文件的目录。具体操作是，单击[Settings]**按钮，设置界面，如下图：**



在“Base Directory” **中输入文件路径。单击**[OK]**按钮确认。**

第三步： 在交换机上执行write**命令保存当前配置到配置文件中去。**

第四步： 将文件备份到PC**上，执行命令**Switch#upload configuration 192.168.0.2 beifen.cfg。

第五步：必要时，将备份文件下载到交换机，执行命令 Switch#download configuration 192.168.0.2 beifen.cfg。

第六步： 要想下载的配置文件能够生效，必须重启交换机，执行命令 Switch#reset。

2.4 软件版本升级

iSpirit 12800系列交换机支持软件版本的在线升级。升级是通过工具TFTP 来完成的。

2.4.1 联合文件

交换机在软件版本升级时使用的是联合文件。联合文件中包含管理模块和所有的接口模块的映像程序，是管理模块和接口模块的映像程序打包而形成的一个文件。用户在升级时只需要使用联合文件,在升级过程中系统会自动地更新管理模块和所有在位的接口模块的映像文件。

2.4.2 软件版本升级的命令

在全局配置模式下升级交换机的联合文件，命令如下：

download union <ip-address> <file-name>

其中<ip-address>为运行TFTP服务器的PC的IP地址，<file-name> 为在TFTP服务器上保存的联合文件名。

在升级的过程中不能断电，否则交换机的联合文件可能损坏而造成交换机启动不了。下载完毕后,需要重新启动交换机才能运行新下载的联合文件程序。整个升级过程需要几分钟，请您耐心等候。

在软件版本的升级过程中，系统会自动更新所有在位的接口模块的映像文件，此时不能重启交换机或断电，否则接口模块中的映像文件可能破坏，造成接口模块下一次启动不了。用户必须等到所有的接口模块都更新完毕后才能重启交换机或断电，在升级过程中在Console终端上有相应的提示。

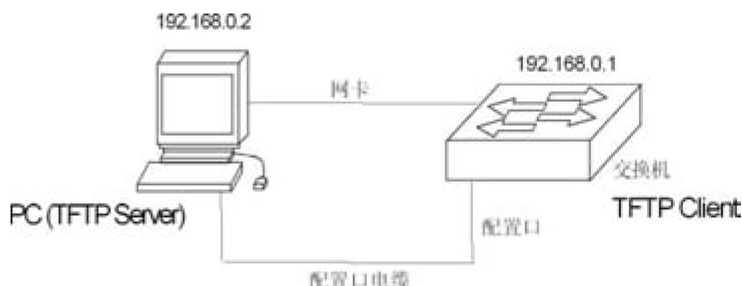
如果使用Telnet终端升级时要特别注意，在Telnet终端上升级完联合文件后会出现CLI提示符，但此时接口模块的映像文件还没有更新，在Telnet终端上也没有任何提示，用户最好在升级完联合文件后再等待几分钟再重启交换机或断电。

建议用户使用Console终端对交换机进行软件版本升级。

2.4.3 软件升级过程

升级联合文件步骤如下：

第一步：搭建升级环境。如下图所示。

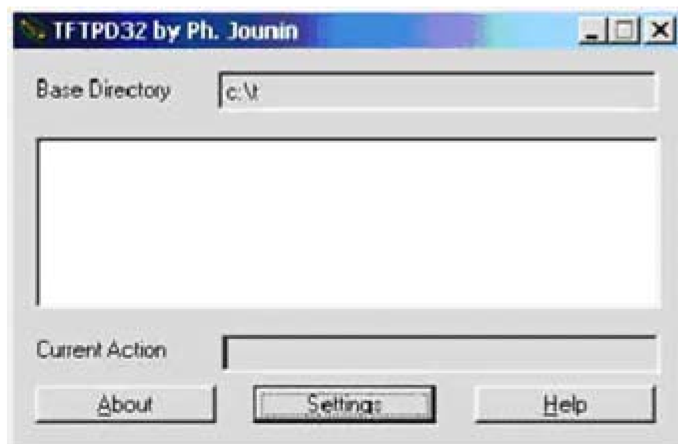


搭建过程如下：

- **将交换机的**Console**口通过电缆外接一台配置终端**（PC）。
- **在**PC**上安装**TFTP Server。
- **将新的联合文件拷贝到**PC**的某一路径下，这里假定路径为**c:\t；
- **配置**PC**的以太网口**IP **地址，这里假定**PC**的**IP**地址为**192.168.0.2 。
- **配置交换机的**IP**地址，这里假定交换机的**IP **地址为**192.168.0.1。

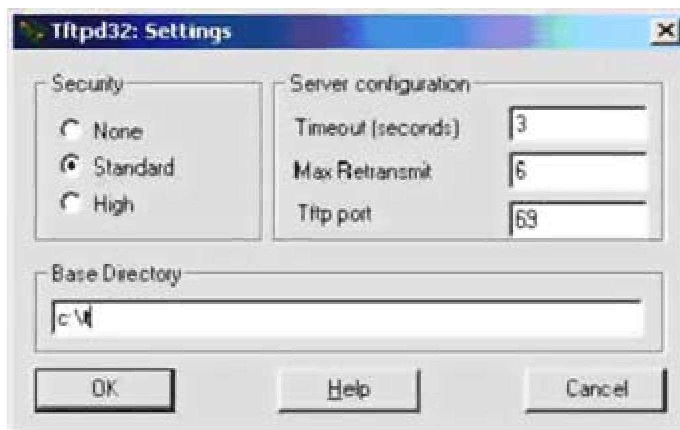
第二步：运行TFTP Server，**并配置**TFTP**服务器。**

首先： **运行**TFTP Server 。TFTPD32 **窗口界面如下图：**



然后： **设置**TFTP Server **文件目录。启动**TFTP Server **之后，重新设置**TFTP Server

文件目录，将待加载的联合文件拷贝到此目录之中。具体操作是，单击[Settings]**按钮，出现**TFTPD32 **设置界面，如下图。**



在"Base Directory" **中输入文件路径。单击**[OK]**按钮确认。**

第三步：升级文件。

首先：将交换机的端口与运行TFTP Server **程序的**PC**通过以太网线连接。并用**ping **命令检测主机与交换机之间是否连通。**

然后：在超级终端 Switch#**提示符下输入命令：**

Switch# download union 192.168.0.2 lenovo.uni**，回车，等待升级联合文件完毕。**

Loading…4136294

I am updating union,

Please wait and don't shut me down……

Updating union has completed !

I am updating master,

Please wait and don't shut me down……

Updating master has completed !

I am updating version,

Please wait and don't shut me down……

Updating version has completed !

I am updating Board(24gt) on slot 1,
Please wait and don't shut me down……

Board(24gt) on the slot 1 update has finished,
You should restart it !
Would you like to restart it ?(Y/N)y

I am updating Board(28xt) on slot 2,
Please wait and don't shut me down……

Board(28xt) on the slot 2 update has finished,
You should restart it !
Would you like to restart it ?(Y/N)y
Switch#

注意：

交换机升级过程中，不能断电。

第四步： 重新启动交换机。

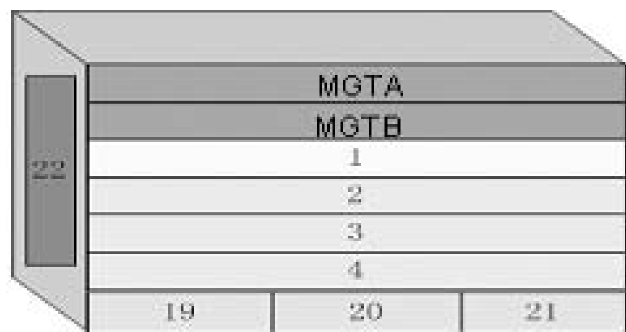
Switch# reset

第3章 配置模块

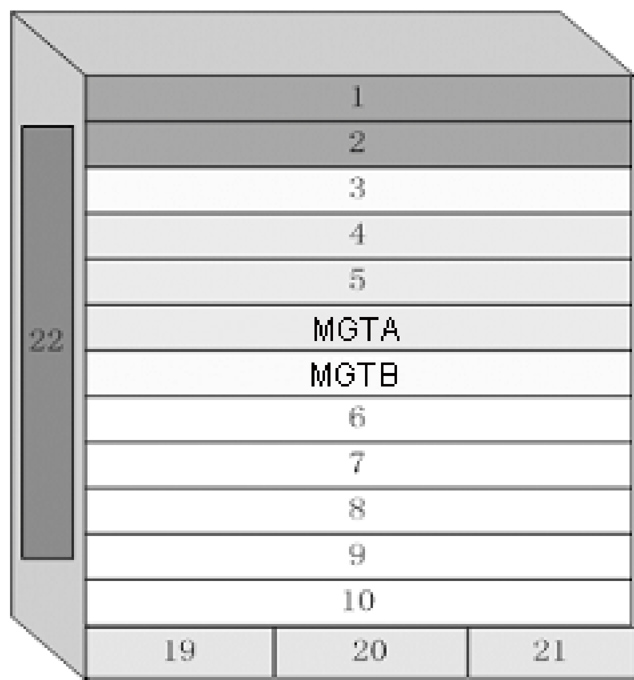
本章对模块相关的配置进行介绍（不包括增值业务模块，比如防火墙模块）。与模块相关的配置主要包括以下内容：

- 管理模块的自动配置
- 业务模块的自动配置
- 业务模块的手动配置
- 电源模块的自动配置
- 风扇模块的自动配置
- 管理模块，业务模块，电源模块，风扇模块信息查询
- 风扇模块故障自动告警
- 电源模块故障自动告警
- 业务模块不匹配自动告警

iSpirit 12800系列交换机包括iSpirit 12804和iSpirit 12810两款产品，均支持管理模块（可插在MGTA 和MGTB槽位），业务模块（iSpirit 12804可插在1 2 3 4槽位，iSpirit 12810可插在1 2 3 4 5 6 7 8槽位），电源模块（可插在19 20 21槽）和风扇模块（可插在22槽位）的配置（iSpirit 12810的9 10槽位为增值业务槽，在此不作说明）。其主要命令是完全兼容的。下面的图示说明了iSpirit 12804和iSpirit 12810的槽位分布。



iSpirit 12804 槽位分布图



iSpirit 12810 槽位分布图

3.1 管理模块的自动配置

交换机上电后，管理模块会自动启动自己的心跳，方便管理员查看管理模块运行是否正常，一般情况下，心跳正常表示管理模块正常，否则表示管理模块出现问题（此时如果存在主备两个管理模块而且当前模块为主用模块，则会发生主备倒换）。管理模块的心跳有两种查看方式：

- 通过模块信息查询命令查看管理模块心跳状态。
- 通过管理模块面板上的sys灯查看模块的心跳状态：sys灯闪烁表示模块心跳正常；sys灯常亮或者常灭表示模块心跳不正常。

交换机上电后，也会自动检测管理模块（包括主用管理模块和备用管理模块）的主备用状态以及状态变化。交换机能够自动判断出两块管理模块中哪个为主用管理模块，哪个为备用管理模块。关于主用管理模块和备用管理模块又分成如下几种情况：

- 只有A槽位插入管理模块时，此管理模块即为主用管理模块
- 只有B槽位插入管理模块时，此管理模块即为主用管理模块
- A槽位和B 槽位均插入管理模块时，先插入的管理模块即为主用管理模块

管理模块的主备用信息有两种查看方式：

- 通过模块信息查询命令查看管理模块主备用状态。
- 通过管理模块面板上的mode灯查看模块的主备用状态，mode灯亮表示此管理模块为主用状态，mode灯灭表示此管理模块处于备用状态。

交换机上电后，同时会自动检测管理模块（包括主用管理模块和备用管理模块）的硬件在线状态。管理员可以通过模块信息查询命令查看A槽和B槽上管理模块是否硬件在位。

3.2 业务模块的自动配置

交换机上电后，能够自动检测到交换机上插入的业务模块。交换机能识别插入的业务模块类别，并自动进行上线处理。业务模块的上线状态分为2个级别：

- 硬件在线（离线）：此种状态表示相应的槽位的硬件是否已经在位准备好
- 软件在线（离线）：此种状态表示相应槽位的业务模块软件是否已经准备好

注意：硬件在线不一定能保证软件在线，但软件在线一定要求硬件在线，下面的表格说明了可能出现的情况：

硬件在线状态	软件在线状态	说明
在线	在线	表示业务模块软硬件均已经准备好，可以进行相应的操作
在线	离线	此种状态可能在以下条件下出现： 业务模块已经插入相应槽位，正在进行软件上线的处理过程 业务模块已经插入相应槽位，但由于业务模块损坏导致软件启动不正常 业务模块已经插入相应槽位，但该槽位原来已经配置了另外一种类型的业务模块(用户手动配置或是从配置文件配置)（此种情况下会同时上报业务模块不匹配告警）。
离线	在线	正常情况下突然拔出业务模块，软件正在下线处理过程中会出现此种状态
离线	离线	业务模块没有插入相应槽位

业务模块的软硬件在线状态可以通过模块信息查询命令获得。

如果相应的槽位并没有插入模块，则手动创建的模块软硬件在线状态均为离线。

业务模块自动上线时，将会通过log**形式通知管理员，其上报内容如下：**

参数名称	参数说明
Alm_type	消息类型，此处为BD_ONLINE ：业务模块上线
Alm_level	消息重要级别，此处为5：很重要
message	其他信息： bd_type：自动配置的业务模块类型 slotid：业务模块槽位号

业务模块上线后，管理员手动拔除业务模块，交换机将会对业务模块自动进行下线处理，同时通过log**形式通知管理员，其上报内容如下：**

参数名称	参数说明
Alm_type	消息类型，此处为BD_OFFLINE：业务模块下线
Alm_level	消息重要级别，此处为5：很重要
message	其他信息： bd_type：自动下线的业务模块类型 slotid：业务模块槽位号

3.3 业务模块的手动配置

业务模块除了支持自动配置外，还支持用户手动进行配置，主要包括：

- 用户手动创建业务模块
- 用户手动删除业务模块

3.3.1 用户手动创建业务模块

下面的命令在模块配置模式下手动创建业务模块：

Switch(config-module)#create module <module-id> <module-type>

例如在槽位1上创建24GT模块：

Switch(config-module)#create module 1 24gt

其中：

槽位号module-id在iSpirit 12804中可以取1 2 3 4，在iSpirit 12810中可以取1 2 3 4 5 6 7 8。

模块类型module-type为当前支持的模块，如下所示，以后还将支持新的模块。

模块名称	模块说明
24gt	24口千兆电口模块
24gx 10g2x	24口千兆光口模块 2口万兆光口模块
28xt	12口千兆光口+16口千兆电口模块
16xt	12口千兆光口+4口千兆电口模块

如果模块已经创建（或自动上线）并且模块类型和现在参数中的模块类型不相同，则返回失败，如果相同或原来槽位并没有创建（或自动上线）模块则返回成功。

3.3.2 用户手动删除业务模块

下面的命令在模块配置模式下手动删除业务模块：

Switch(config-module)#remove module <module-id>

例如删除槽位1**上创建的**24GT**模块：**

Switch(config-module)#remove module 1

其中：

槽位号module-id**在**iSpirit 12804**中可以取**1 2 3 4**，在**iSpirit 12810**中可以取**1 2 3 4 5 6 7 8。

如果相应的槽位并没有模块，则返回失败。

如果相应的槽位存在模块而且硬件软件在线，删除后过一段时间此模块又会自动创建并自动上线。

3.4 电源模块的自动配置

交换机上电后，交换机会启动对电源模块的检测，并在电源出现故障时上报告警。电源状态的信息有两种方式获得：

- **通过模块信息查询命令获得电源模块在位，运行是否正常等状态。**
- **通过面板上的**PWR1 PWR2 PWR3**灯查看三个电源的运行状态。灯常亮表示电源在线且运行正常，灯闪烁表示电源在线但运行不正常，灯不亮表示电源不在线。**

3.5 风扇模块的自动配置

交换机上电后，交换机会启动对风扇模块的检测，并在风扇出现故障时上报告警。风扇状态的信息有两种方式获得：

- **通过模块信息查询命令获得风扇模块在位，运行是否正常等状态。**

- **通过面板上的**FAN**灯查看风扇的运行状态。灯常亮表示风扇在线且运行正常，灯闪烁表示风扇在线但运行不正常，灯不亮表示风扇不在线。**

3.6 管理模块，业务模块，电源模块，风扇模块信息查询

下面的命令在普通模式或特权模式下查询管理模块，业务模块，电源模块，风扇模块的运行状态：

Switch#show module [module-id]

例如查询槽位1**上的业务模块信息如下：**

Switch#show module 1

slotid	modulename	online	other
1	24gt	online	on /heartbeat

其中：槽位号module-id**在**iSpirit 12804**中可以取**1 2 3 4**（业务模块）**17 18**（**MGTA**和**MGTB**模块）**19 20 21**（电源模块）**22**（风扇模块）；在**iSpirit 12810**中可以取**1 2 3 4 5 6 7 8**（业务模块）**17 18**（**MGTA**和**MGTB**模块）**19 20 21**（电源模块）**22**（风扇模块）。**

也可以查询所有模块的信息，只要不输入槽位号module-id**参数就可以了，如下：**

Switch#show module

slotid	modulename	online	other
1	24gt	online	on /heartbeat
2	null	offline	off/no heartbeat
3	null	offline	off/no heartbeat
4	null	offline	off/no heartbeat
17	mgt-a	online	master
18	mgt-b	offline	slave /no heartbeat
19	power	offline	
20	power	online	
21	power	offline	
22	fan	online	fail

查询返回的各参数说明如下：

参数名称	参数说明
slotid	槽位号
modulename	模块名称： 24gt：24口千兆电口板 24gx：24口千兆光口板 10g2x：2口万兆光口板 28xt：12口千兆光口+16口千兆电口模块 16xt：12口千兆光口+4口千兆电口模块 mgt-a：主控A槽 mgt-b：主控B槽 power：电源板 fan：风扇板
online	软件在线标志
other	业务模块：业务模块硬件是否在线/业务模块心跳是否正常 管理模块：管理模块主用备用状态/管理模块心跳是否正常 电源风扇：运行是否正常，fail表示运行不正常；warning表示存在告警

3.7 风扇模块故障自动告警

交换机上电后，交换机会启动对风扇模块的检测，当风扇模块出现故障时，交换机会自动上报告警，其内容包括：

参数名称	参数说明
Alm_type	告警类型，此处为FAN_WARN：风扇告警
Alm_level	告警级别，此处为5：严重故障

message	**其他消息：** slotid**：模块槽位号，此处固定为**22 brd_type**：模块类型类型，此处固定为**fan status**：板状态，在线/离线，有告警时板状态後将有**warning**显示**

交换机启动时风扇故障则上报告警一次，同时风扇由好变坏或由坏变好时上报告警。

3.8 电源模块故障自动告警

交换机上电后，交换机会启动对电源模块的检测，当电源模块出现故障时，交换机会自动上报告警，其内容包括：

参数名称	**参数说明**
Alm_type	**告警类型，此处为**POWER_WARN**：电源告警**
Alm_level	**告警级别，此处为**5**：严重故障**
Message	**其他消息：** slotid**：电源模块槽位号** brd_type**：模块类型类型，此处为**power status**：板状态，在线/离线，有告警时板状态後将有**warning**显示**

交换机启动时电源故障则上报告警一次,同时任何一个电源模块由好变坏或由坏变好时上报告警。

3.9 业务模块不匹配自动告警

如果管理员手动创建了一个离线模块，然后在相应槽位插入了一个不匹配的模块，则交换机将上报业务模块不匹配告警。告警内容如下所示：

参数名称	参数说明
Alm_type	告警类型，此处为BD_NOMATCH：业务模块不匹配
Alm_level	告警级别，此处为5：严重故障
message	其他消息： slotid：业务模块槽位号 old_type：原来创建的业务模块类型 new_type：硬件实际插入的业务模块类型

发生此种告警时，此告警将每隔2秒一直上报，直到管理员干预，管理员可以拔除不匹配的业务模块或手动删除原来创建的业务模块来消除此告警。

第4章 配置端口

本章对端口相关的配置进行介绍，主要包括以下内容：

- **端口的通用配置**
- **配置**MIRROR
- **配置**STORM-CONTROL
- **配置**FLOW-CONTROL
- **配置端口带宽**

4.1 端口的通用配置

管理员通过对交换机的端口配置控制端口下接入的用户，如不让端口下的用户接入网络，管理员可以关闭这个端口。本节对端口的通用配置进行介绍，主要包括：

- 端口的打开和关闭
- 端口的速率配置
- 显示端口的信息

4.1.1 端口的打开和关闭

iSpirit 12800系列交换机的端口缺省是打开的，如果管理员希望端口下的用户不能接入网络，可以关闭这个端口。

下面的命令在接口配置模式下打开端口的管理状态：

no shutdown

例如打开端口1/1的管理状态：

Switch(config-ge1/1)#no shutdown

下面的命令在接口配置模式下关闭端口的管理状态：

Shutdown

例如关闭端口1/1的管理状态：

Switch(config-ge1/1)#shutdown

4.1.2 端口的速率配置

所有的端口的缺省速率配置是自适应（autonegotiate）的。

下面的命令在接口配置模式下配置端口的速率：

bandwidth <bandwidth-value>

例如端口1/1的速率配置成1000M：

Switch(config-ge1/1)# bandwidth 1000m

对于不同不同类型的端口，其可以设置的值也不相同，如下表所示：

端口类型	速率
千兆电口	10m
	100m
	1000m
千兆光口	1000m
万兆光口	10000m

注意：m**表示兆，**k**表示千，**g**表示千兆，所以**1000m**可以写成**1g**或者**1000000k

另外还有一个命令在接口配置模式下配置端口的双工模式：

duplex {full | half | auto}

例如端口1/1**的速率配置成强制全双工：**

Switch(config-ge1/1)#duplex full

对于不同类型的端口，可以设置的模式也不相同，如下表所示：

端口类型	模式
千兆电口	Full
	Half
	Auto
千兆光口	Full
	Auto
万兆光口	Full
	Auto

4.1.3 显示端口的信息

下面的命令在普通模式或特权模式下显示一个或多个端口的信息：

show interface [if-name]

例如显示端口1/1**的信息：**

Switch# show interface ge1/1

例如显示所有端口的信息：

Switch# show interface

4.2 配置 MIRROR

端口镜像对于监听一个或多个端口接收和发送的包的流量是一个非常有用的功能，它能用镜像端口去监听一个或多个端口的接收和发送的包。联想天工iSpirit 12800系列交换机支持端口镜像功能，镜像端口能够监听别的端口的进入的数据和出去的数据。一个镜像端口可以同时监听多个端口。本节重点介绍MIRROR 的配置，主要包括以下内容：

- 配置MIRROR的监听端口和被监听端口
- 显示MIRROR配置

4.2.1 配置 MIRROR 的监听端口和被监听端口

管理员配置监听端口的时候，需要进入此接口配置模式设置被监听端口，例如设置端口ge1/1监听端口ge1/2，则需要进入端口ge1/1下，键入命令：

Switch(config-ge1/1)# mirror interface ge1/2 direction both

此时，端口ge1/1被设置为监听端口，ge1/2被设置为被监听端口。

设置被监听端口的命令如下：

Switch(config-ge1/1)#mirror interface <if-name> direction {both | receive | transmit}

此时，端口ge1/1设置为监听端口，<if-name> 设置为被监听端口，同时后面的{both | receive | transmit}指明了监听的方向：receive表示监听收到的数据包；transmit监听发送的数据包；both监听发送和接收的所有数据包。比如：

Switch(config-ge1/1)#mirror interface ge1/2 direction both

表示设置端口ge1/1监听端口ge1/2的发送和接收的数据包。

如果要设置多个被监听端口，需要执行多次命令。

管理员在接口配置模式下，可以取消被监听端口，命令如下：

Switch(config-ge1/1)#no mirror interface <if-name>

此时<if-name>是不再被监听的端口。比如：

Switch(config-ge1/1)# no mirror interface ge1/2

表示设置端口ge1/1不再监听端口ge1/2的数据包。

当所有被监听端口都被取消时，监听端口也将被清除。

4.2.2 显示 MIRROR 的配置

管理员可以在普通模式或特权模式下通过下面命令查看已经设置的MIRROR**配置：**

Switch# show mirror

需要注意以下几点：

- **一个端口不能同时设置为监听端口和被监听端口。**
- **监听端口只能有一个，但被监听端口可以有多个。**
- **交换机支持跨模块的端口监听功能。**

4.3 配置 STORM-CONTROL

在现实生活中，一个NIC**卡发很高速率的单播、组播、广播包可以使得网络出现故障，在这种情况下，交换机上的抑制功能便显得尤为重要，它能防止数据包涌进网络而造成网络拥塞的情况，联想天工**iSpirit 12800**系列交换机的所有端口支持广播包、组播包和**DLF **包的抑制功能。**

本节对STORM-CONTROL**的配置进行详细的描述，主要包括以下内容：**

- **缺省配置**
- **广播抑制配置**
- **组播抑制配置**
- DLF **抑制配置**
- **显示**STORM-CONTROL**配置**

4.3.1 缺省配置

iSpirit 12800**系列交换机支持对每个端口分别设置**broadcast rate, multicast rate, dlf rate **。默认端口的广播包抑制到**1500**个，目的是防止网络形成广播风暴。**DLF **包缺省抑制到**1500**个包，组播包缺省没有做抑制。**

4.3.2 广播抑制配置

下面的命令在接口配置模式下配置此端口的广播抑制：

storm -control broadcast level <rate>

其中rate**的范围为**0.00–100.00**，表示百分比。**

下面的命令在接口配置模式下取消此端口的广播抑制的配置：

no storm-control broadcast level

4.3.3 组播抑制配置

下面的命令在接口配置模式下配置此端口的组播抑制：

storm -control multicast level <rate>

其中rate**的范围为**0.00–100.00**，表示百分比。**

下面的命令在接口配置模式下取消此端口的组播抑制的配置：

no storm-control multicast level

4.3.4 DLF 抑制配置

下面的命令在接口配置模式下配置此端口的DLF **抑制：**

storm -control dlf level <rate>

其中rate**的范围为**0.00–100.00**，表示百分比。**

下面的命令在接口配置模式下取消此端口的DLF **抑制的配置：**

no storm-control dlf level

4.3.5 显示 STORM-CONTROL 配置

下面的命令在普通模式或特权模式下显示STORM-CONTROL**配置：**

show storm-control

4.4 配置 FLOW-CONTROL

FLOW-CONTROL（流量控制）用于防止在端口阻塞的情况下数据丢包。在半双工方式下，流量控制通过背压（Backpressure）技术实现，使得信息源降低发送速率。在全双工模式下，流量控制遵循IEEE802.3x 标准，阻塞端口向信息源发送"Pause"包令其暂停发送。

本节对FLOW-CONTROL（流量控制） 的配置进行详细的描述，主要包括以下内容：

- 缺省配置
- 设置端口发送侧流控
- 设置端口接收侧流控
- 关闭端口流控
- 显示流控信息

4.4.1 缺省配置

iSpirit 12800系列交换机支持对每个端口分别设置发送和接收的流控。默认端口并没有打开流控功能。

4.4.2 设置端口发送侧流控

下面的命令在接口配置模式下配置端口发送侧流控打开：

flowcontrol send on

下面的命令在接口配置模式下配置端口发送侧流控关闭：

flowcontrol send off

4.4.3 设置端口接收侧流控

下面的命令在接口配置模式下配置端口接收侧流控打开：

flowcontrol receive on

下面的命令在接口配置模式下配置端口接收侧流控关闭：

flowcontrol receive off

4.4.4 关闭端口流控

下面的命令在接口配置模式下关闭端口发送和接收侧流控：

no flowcontrol

4.4.5 显示流控信息

下面的命令在普通模式或特权模式下显示所有端口的流控信息：

show flowcontrol

下面的命令在普通模式或特权模式下显示某一个端口的流控信息：

show flowcontrol interface <if-name>

其中，<if-name>**为要查询流控信息的端口名称。**

4.5 配置端口带宽

端口带宽控制用于控制端口发送和接收的速率。

本节对端口带宽的配置进行详细的描述，主要包括以下内容：

- **缺省配置**
- **设置端口发送或接收带宽控制**
- **取消端口发送或接收带宽控制**
- **显示端口配置的带宽控制**

4.5.1 缺省配置

iSpirit 12800**交换机支持对每个端口分别设置发送和接收的带宽。默认端口并没有进行**

带宽控制。

4.5.2 设置端口发送或接收带宽控制

下面的命令在接口配置模式下设置端口发送或接收带宽控制：

portrate {egress | ingress} <rate>

egress**表示对发送的数据包进行带宽控制。**

ingress**表示对接收的数据包进行带宽控制。**

<rate>**表示要设置的带宽的值，范围为**1－1048512**，单位为**kbits**。**

4.5.3 取消端口发送或接收带宽控制

下面的命令在接口配置模式下取消端口的带宽控制：

no portrate {egress | ingress}

egress**表示取消发送数据包的带宽控制。**

ingress**表示取消接收数据包的带宽控制。**

4.5.4 显示端口配置的带宽控制

下面的命令在普通模式或特权模式下查看端口配置的带宽控制：

show portrate interface <if-name>

其中<if-name>**为要查询带宽控制信息的端口名称。**

第5章 配置VLAN

VLAN 是交换机中的一个重要概念，在实际应用中使用非常多，它是内部划分多个网络的基础。VLAN 是虚拟局域网的简称，它是逻辑地把多个设备组织在一起的一个网络，而不管设备的物理位置在哪里。每个VLAN 都是一个逻辑网络，它具有传统的物理网络的一切功能和属性。每个VLAN都是一个广播域，广播包只能在一个VLAN 内进行转发，不能跨越VLAN，VLAN间的数据通信必须通过三层转发。

本章主要包括以下内容：

- VLAN 介绍
- VLAN 配置
- VLAN 配置示例

5.1 VLAN 介绍

本节对VLAN 进行一个详细的介绍，主要包括以下内容：

- VLAN 的好处
- VLAN ID
- VLAN 端口成员类型
- 端口的缺省VLAN
- 端口的VLAN模式
- VLAN 中继
- 数据流在VLAN 内的转发
- VLAN 的子网

5.1.1 VLAN 的好处

VLAN 极大地扩展了物理网络的规模。传统的物理网络只能有一个很小的规模，最多能容纳上千台设备，而使用VLAN 划分的物理网络能够容纳上万甚至几十万台设备。VLAN 与传统的物理网络有相同的功能和属性。

使用VLAN 有以下好处：

- VLAN 能有效控制网络中的流量。

在传统网络中，不管有无必要，所有的广播包都传送到所有的设备，加重了网络和设备的负载。而VLAN 能够根据需要把设备组织在一个逻辑网络中，一个VLAN 就是一个广播域，广播包只在VLAN 内部传送，不会跨越VLAN 。通过划分VLAN 可以有效地控制网络中的流量。

- VLAN 能够提高网络的安全性。

VLAN 内的设备只能与同一个VLAN 的设备进行二层通信，如果要与另一个VLAN 通信，必须通过三层转发，如果不建立VLAN 间的三层转发，VLAN 间完全不能通信，可以起到隔离的作用，保证每个VLAN 内的数据安全。例如一个公司研发部不想与市场部的数据进行共享，可以研发部建立一个VLAN， 市场部建立一个VLAN， 二个VLAN 间不建立三层通信通道。

- VLAN 使设备的移动变得方便。

传统的网络中的设备如果从一个位置移动到另一个位置而属于不同的网络时，需要修改移动

设备的网络配置，这样对于用户来说是非常不方便的。而VLAN 是一个逻辑网络，可以把不在同一物理位置的设备划在同一个网络，当设备移动时还可以使设备属于此VLAN 中，这样移动的设备不需要修改任何配置。

5.1.2 VLAN ID

每一个VLAN 有一个标识号，叫VLAN ID，VLAN ID 的范围从0 到4095，其中0 和4095 不用，实际有效的只有1 到4094 。VLAN ID 唯一标识一个VLAN 。

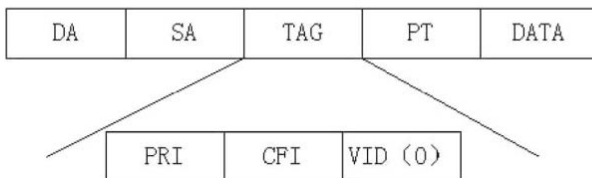
iSpirit 12800系列交换机支持4094 个VLAN ，在创建VLAN 时，要选择一个VLAN ID ，范围从2到 4094。交换机在缺省情况下创建了VLAN1，并且VLAN1是不能被删除的。

在网络中的一个VLAN 内传输的数据帧有三种： 不带标记的数据帧，带VID 为0 的标记的数据帧，带VID 非0 的标记的数据帧。如下图所示为三种不同数据帧格式。

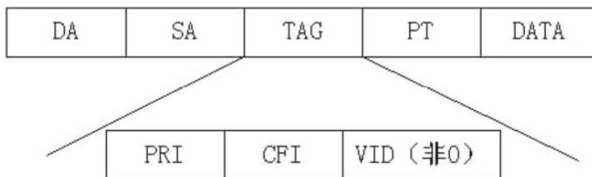
不带标记的数据帧



带标记的数据帧，但VLAN ID为0



带标记的数据帧，但VLAN ID非0



在交换机内部所有的数据帧都是带标记的。如果一个不带标记的数据帧输入交换机，交换机要给该数据帧加上一个标记，选择一个VLAN ID 值填入标记的VID 中。如果一个带VID 为0 的标记的数据帧输入交换机，交换机选择一个VLAN ID 值填入标记的VID 中。如果一个带VID 非0 的标记的数据帧输入交换机，该帧不变。

5.1.3 VLAN 端口成员类型

交换机支持基于端口的VLAN 和基于802.1Q 的VLAN 。一个VLAN 包括两种端口成员类型：untagged 成员和tagged 成员。一个VLAN 可以既包括untagged 端口成员，又包括tagged 端口成员。

一个VLAN 可以没有端口成员，也可以有一个或多个端口成员。当一个端口属于一个VLAN 时，可以是VLAN 的untagged 成员或tagged 成员。

一个端口可以属于一个或多个VLAN 的tagged或untagged成员，如果一个端口属于两个或多个VLAN 的tagged成员时，这个端口又称为VLAN 中继端口。一个端口可以同时属于一个或多个VLAN 的untagged 成员和属于另外的一个或多个VLAN 的tagged 成员。

5.1.4 端口的缺省 VLAN

端口有且只有一个缺省VLAN，缺省VLAN用于决定从该端口输入的不带标记或带标记但VID为0的数据包的所属VLAN。缺省VLAN又被称为端口VID或PVID。缺省情况下，端口的缺省VLAN为1。

5.1.5 端口的 VLAN 模式

端口存在三种VLAN模式：ACCESS模式，TRUNK模式和HYBRID模式。用户进行端口的VLAN配置时必须首先指定端口的VLAN模式。

ACCESS模式的端口是一个接入端口，直接面向用户，该端口只能属于一个VLAN的untagged成员，缺省VLAN是用户指定的VLAN。当端口只属于一个VLAN的untagged成员时，可以指定该端口的VLAN模式为ACCESS模式。

TRUNK模式的端口是一个中继端口，直接与交换机相连，该端口可以属于一个或多个VLAN的tagged成员，但不能属于任何VLAN的untagged成员，该端口的缺省VLAN为1，不能改变。

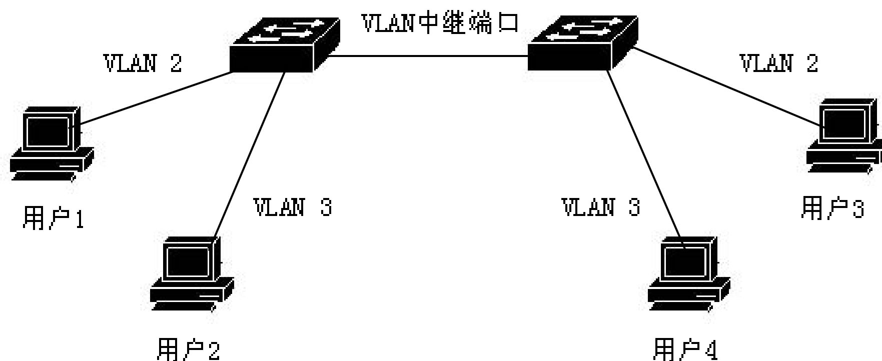
HYBRID模式的端口是一个中继端口，直接与交换机相连，该端口可以属于一个或多个VLAN的tagged成员和（或）一个或多个VLAN的untagged成员。该端口的缺省VLAN可以改变。

在实际应用时，用户可以根据具体情况来选择端口的VLAN模式。

5.1.6 VLAN 中继

如果一个端口属于两个或多个VLA N 的tagged成员，那么这个端口又称为VLAN 中继端口。两个交换机之间可以以VLAN 中继端口相连，这样两个交换机之间可以划分两个或多个共同的VLAN 。

如下图 是一个VLAN 中继的例子，两个交换机之间以VLAN 中继端口相连，是VLAN 2和VLAN 3 的中继端口，每个交换机划分为两个VLAN， 分别是VLAN 2 和VLAN 3， 每个VLAN 内有一个用户。这样，用户1 可以与用户3 通信，用户2 可以与用户4 通信，用户1 和用户3 不能与用户2 和用户4 通信。



5.1.7 数据流在 VLAN 内的转发

当交换机从一个端口收到一个数据包时，根据以下步骤进行二层转发：

- 决定该数据包所属的VLAN 。
- 判断该数据包是广播数据包、组播数据包还是单播数据包。
- 根据不同的数据包确定输出端口（可以是零个、一个或多个输出端口）， 如果没有输出端口，丢弃该数据包。
- 根据输出端口在VLAN 内的成员类型决定发出去的包是否带标记。
- 从输出端口发送出去。

1) 如何决定数据包的所属 VLAN：

如果收到的数据包带标记并且标记中的VID **字段非**0**时，该数据包所属的**VLAN **就是标记中**V I D **值。**

如果收到的数据包不带标记或带标记但标记中的VID **值为**0**时，该数据包所属的**VLAN**是端口的缺省**VLAN**。**

2) 如何确定数据包的类型：

如果收到的数据包的目的MAC **地址是**FF:FF:FF:FF:FF:FF**，则该数据包是广播数据包。**

如果收到的数据包不是广播数据包且其目的MAC **地址的第**40 **位为**1**，则该数据包是组播数据包。**

如果既不是广播数据包又不是组播数据包，则该数据包为单播数据包。

3) 如何决定数据包的输出端口：

如果输入的数据包是广播数据包，该数据包所属的VLAN **的所有成员端口就是数据包的输出端口。**

如果输入的数据包是组播数据包，首先根据目的组播MAC **地址和所属的**VLAN **查找二层硬件组播转发表，如果找到匹配的组播条目，则组播条目中的输出端口和所属**VLAN **中的成员端口中的共同端口（与操作）为数据包的输出端口，如果没有共同的端口，该数据包丢弃。如果在二层硬件组播转发表中没有找到匹配的组播条目，根据二层硬件组播转发表的转发模式决定输出端口，如果是未注册组播转发模式，组播包当作广播处理，所属的**VLAN **的所有成员端口就是数据包的输出端口，如果是注册转发模式，则没有输出端口，数据包丢弃。**

如果输入的数据包是单播数据包，首先根据目的MAC **地址和所属的**VLAN **查找二层硬件转发表，如果找到匹配的条目，则条目中的输出端口与所属**VLAN **的成员端口中的共同端口（与操作）为数据包的输出端口，如果没有共同的端口，该数据包丢弃。如果在二层硬件转发表中没有找到匹配的条目，该数据包当作广播包处理，所属的**VLAN **的所有成员端口就是数据包的输出端口。**

4) 发送数据包：

决定了输入的数据包的输出端口后要把数据包从所有的输出端口发送出去。

如果某个输出端口是数据包所属的VLAN **的**untagged **成员，则数据包从该输出端口发送出去时不带标记。**

如果某个输出端口是数据包所属的VLAN **的**tagged **成员，则数据包从该输出端口发送**

出去时带标记，标记中的VID 值是数据包所属的VLAN 的值。

5.1.8 VLAN 的子网

在交换机上一个VLAN 是一个广播域， 一个VLAN 上可以建立一个子网接口，所有的子网都是建立在VLAN 的基础上的。iSpirit 12800系列交换机上最多可划分4094 个VLAN，但最多只能建立512 个子网，当在512 个VLAN 上建立了子网后，其它的VLAN 就不能建立子网接口。子网接口的创建和删除不需要用户的干预，是系统自动完成的，当用户创建一个VLAN时，该VLAN对应的子网接口自动建立，当用户删除一个VLAN时，该VLAN对应的子网接口也被删除。

5.2 VLAN 配置

本节对VLAN 的配置进行详细的介绍，主要包括以下内容：

- 创建和删除VLAN
- 配置端口的VLAN模式
- ACCESS模式的VLAN配置
- TRUNK 模式的VLAN配置
- HYBRID模式的VLAN配置
- 查看VLAN的信息

5.2.1 创建和删除 VLAN

在创建和删除VLAN之前，用户需要在全局配置模式下使用vlan database命令进入VLAN配置模式，在该模式下创建和删除VLAN。

系统在缺省情况下已经创建了VLAN 1，并且VLAN 1不能被用户删除。创建和删除VLAN的命令如下表：

命令	描述	CLI模式

vlan <vlan-id>	创建一个VLAN。如果该VLAN已经存在，则不做处理，否则创建此VLAN。参数的范围从2到4094。	VLAN配置模式
no vlan <vlan-id>	删除一个VLAN，如果该VLAN不存在，则不做处理，否则删除此VLAN。参数的范围从2到4094。	VLAN配置模式

5.2.2 配置端口的 VLAN 模式

在配置端口的VLAN之前需要指定端口的VLAN模式，缺省情况下端口的VLAN模式是ACCESS模式。指定端口的VLAN模式的命令如下表：

命令	描述	CLI模式
switchport mode access	指定端口的VLAN模式是ACCESS模式。执行此命令后端口是VLAN1的untagged成员，端口的缺省VLAN是1。	接口配置模式
switchport mode trunk	指定端口的VLAN模式是TRUNK模式。执行此命令后端口是VLAN1的tagged成员，端口的缺省VLAN是1。	接口配置模式
no switchport trunk	端口的VLAN模式不再是TRUNK模式，回到缺省的情况，即ACCESS模式。	接口配置模式
switchport mode hybrid	指定端口的VLAN模式是HYBRID模式。执行此命令后端口是VLAN1的untagged成员，端口的缺省VLAN是1。	接口配置模式
no switchport hybrid	端口的VLAN模式不再是	接口配置模式

	HYBRID模式，回到缺省的情况，即ACCESS模式。	

5.2.3 ACCESS 模式的 VLAN 配置

端口做VLAN配置之前需要指定端口的VLAN模式为ACCESS模式。在这种VLAN模式下端口缺省是VLAN1的untagged成员，端口的缺省VLAN是1。ACCESS模式的VLAN配置命令如下表：

命令	描述	CL模式
switchport access vlan <vlan-id>	配置端口是指定的VLAN的untagged成员，端口的缺省VLAN是指定的VLAN。参数范围从2到4094。	接口配置模式
no switchport access vlan	端口的VLAN配置回到缺省情况，即端口是VLAN1的untagged成员，端口的缺省VLAN是1。	接口配置模式

5.2.4 TRUNK 模式的 VLAN 配置

端口做VLAN配置之前需要指定端口的VLAN模式为TRUNK模式。在这种VLAN模式下端口缺省是VLAN1的tagged成员，端口的缺省VLAN是1。TRUNK模式的VLAN配置命令如下表：

命令	描述	CLI模式
switchport trunk allowed vlan all	配置端口是所有VLAN的tagged成员，对于以后新创建的VLAN，该端口也是这些VLAN的tagged成员。	接口配置模式
switchport trunk allowed vlan none	除VLAN1外，该端口不再是	接口配置模式

	所有的其它VLAN的tagged成员。	
switchport trunk allowed vlan add <vlan-list>	配置端口成为指定的一个或多个VLAN的tagged成员。参数 <vlan-list> 可为一个VLAN、一个VLAN范围或多个VLAN。例如参数可以为"1"、"2-4"或"1,3,5"。	接口配置模式
switchport trunk allowed vlan remove <vlan-list>	把端口从指定的一个或多个VLAN中清除，不再是这些VLAN的tagged成员。参数<vlan-list>可为一个VLAN、一个VLAN范围或多个VLAN。例如参数可以为"1"、"2-4"或"1,3,5"。	接口配置模式

5.2.5 HYBRID 模式的 VLAN 配置

端口做VLAN配置之前需要指定端口的VLAN模式为HYBRID模式。在这种VLAN模式下端口缺省是VLAN1的untagged成员，端口的缺省VLAN是1。HYBRID模式的VLAN配置命令如下表：

命令	描述	CLI模式
switchport hybrid vlan <vlan-id>	配置端口是指定的VLAN的untagged成员并且端口的缺省VLAN是指定的VLAN。参数范围从2到4094。	接口配置模式
no switchport hybrid vlan	把端口从缺省VLAN中清除，不再是缺省VLAN的tagged或untagged成员，端口的缺省VLAN回到1。	接口配置模式

switchport hybrid allowed vlan all	配置端口是所有VLAN（VLAN1除外）的tagged成员，对于以后新创建的VLAN，该端口也是这些VLAN的tagged成员。	接口配置模式
switchport hybrid allowed vlan none	除VLAN1外，该端口不再是所有的其它VLAN的tagged或untagged成员，端口的缺省VLAN回到1。	接口配置模式
switchport hybrid allowed vlan add <vlan-list> egress-tagged enable	配置端口成为指定的一个或多个VLAN的tagged成员。参数<vlan-list>可为一个VLAN、一个VLAN范围或多个VLAN。例如参数可以为“1”、“2-4”或“1,3,5”。	接口配置模式
switchport hybrid allowed vlan add <vlan-list> egress-tagged disable	配置端口成为指定的一个或多个VLAN的untagged成员。参数<vlan-list>可为一个VLAN、一个VLAN范围或多个VLAN。例如参数可以为“1”、“2-4”或“1,3,5”。	接口配置模式
switchport hybrid allowed vlan remove <vlan-list>	把端口从指定的一个或多个VLAN中清除，不再是这些VLAN的tagged或untagged成员。如果端口的缺省VLAN属于指定的VLAN，则缺省VLAN回到1。	接口配置模式

5.2.6 查看 VLAN 的信息

查看VLAN的信息的命令如下表：

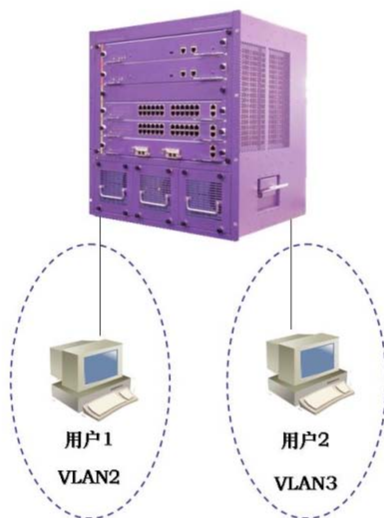
命令	描述	CLI模式
show vlan [vlan-id]	如果不输入参数，显示所有的VLAN信息，如果输入参数，显示指定的一个VLAN信息。参数范围从1到4094。	普通模式，特权模式
show interface switchport	显示系统的所有端口的VLAN相关信息，如VLAN模式，缺省VLAN等。	普通模式，特权模式
show running-config	查看系统当前配置，可以查看到VLAN的配置。	特权模式

5.3 VLAN 配置示例

5.3.1 基于 PORT 的 VLAN

1) 配置

有两个用户，用户1和用户2， 两个用户由于所使用的网络功能和环境不同，需要分别处于不同的VLAN中。用户1属于VLAN2，连接交换机的端口ge1/1，用户2属于VLAN3，连接交换机的端口ge1/2。



交换机的配置如下：

创建VLAN

Switch#config t

Switch(config)#vlan database

Switch(config-vlan)#vlan 2

Switch(config-vlan)#vlan 3

将端口分配到VLAN**中**

Switch#config t

Switch(config)#interface ge1/1

Switch(config-ge1/1)#swithport mode access

Switch(config-ge1/1)#switchport access vlan 2

Switch(config-ge1/1)#exit

Switch(config)#interface ge1/2

Switch(config-ge1/2)#swithport mode access

Switch(config-ge1/2)#switchport access vlan 3

2) 排错

如果配置后，发现不同VLAN **之间的**PC **机不能通信，那是正常现象，因为不同**VLAN **之间要进行通信，必须要经过三层的路由转发。如果同一**VLAN **内的**PC **机不能进行通信，须作以下验证：**

show vlan

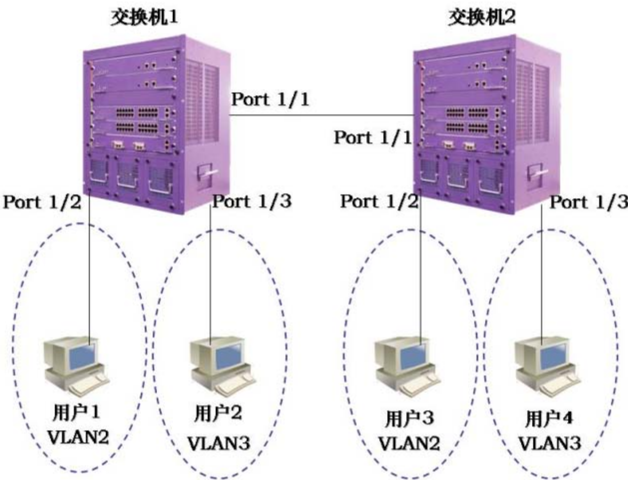
查看所有的VLAN**的成员端口情况**

show vlan <vlan-id>

查看连接特定PC机的端口是否在指定的VLAN内

5.3.2 基于 802.1Q 的 VLAN

1) 配置



有两台交换机分别连接两个用户：

用户	所属VLAN	连接端口	所属交换机	级联端口
用户1	2	1/2	交换机1	1/1
用户2	3	1/3	交换机1	1/1
用户3	2	1/2	交换机2	1/1
用户4	3	1/3	交换机2	1/1

需要在两台交换机上做配置。

交换机1配置：

```
Switch#config t
Switch(config)#vlan database
Switch(config-vlan)#vlan 2
Switch(config-vlan)#vlan 3
Switch#config t
```

```
Switch(config)#interface ge1/1
Switch(config-ge1/1)#switchport mode trunk
Switch(config-ge1/1)#switchport trunk allowed vlan add 2
Switch(config-ge1/1)#switchport trunk allowed vlan add 3
Switch(config)#interface ge1/2
Switch(config-ge1/2)#swithport mode access
Switch(config-ge1/2)#switchport access vlan 2
Switch(config-ge1/2)#exit
Switch(config)#interface ge1/3
Switch(confi g-ge1/3)#swithport mode access
Switch(config-ge1/3)#switchport access vlan 3
```

交换机2配置：

```
Switch#config t
Switch(config)#vlan database
Switch(config-vlan)#vlan 2
Switch(config-vlan)#vlan 3
Switch#config t
Switch(config)#interface ge1/1
Switch(config-ge1/1)#switchport mode trunk
Switch(config-ge1/1)#switchport trunk allowed vlan add 2
Switch(config-ge1/1)#switchport trunk allowed vlan add 3
Switch(config)#interface ge1/2
Switch(config-ge1/2)#swithport mode access
Switch(config-ge1/2)#switchport access vlan 2
Switch(config-ge1/2)#exit
Switch(config)#interface ge1/3
Switch(config-ge1/3)#swithport mode access
Switch(config-ge1/3)#switchport access vlan 3
```

2) 排错

跨交换机的vlan，在同一个vlan内的pc机都能够通信的，如果不能。须查看如下：

- **连接pc 机的端口是否属于相应的VLAN，且应用ACCESS模式加入这个vlan 的。**
- **级联端口1/1是加入到每一个vlan中的，并且端口1/1是TRUNK模式。**

第6章 配置MSTP

本章对MSTP **及其配置进行描述，主要包括以下内容：**

- MSTP **介绍**
- MSTP **配置**
- MSTP **配置示例**

6.1 MSTP 介绍

联想天工iSpirit 12800系列交换机支持IEEE802.1d,IEEE802.1w，IEEE802.1s标准的STP 协议。

6.1.1 概述

MSTP,使用RSTP快速收敛，使多个VLAN聚合到一个生成树实例，每个实例有个一个独立于其他生成树实例的生成树拓扑。这个架构为数据流提供多个转发路径，能够负载均衡，并且减少被要求支持大量VLAN的生成树实例。

6.1.2 多生成树域

对于参与多生成树（MST）计算的实例，必须一致地配置交换机相同的MST配置信息。有相同MST配置的相连的交换机集合构成MST域。

MST 配置决定每一个交换机所属的域。配置包括域名，修订版本号，和 MST 实例和 VLAN 指派映射；这些信息在 MST 配置中会生成一个唯一的摘要（Digest）。同一个域中的摘要是相同的，也必须是相同的，可以通过 show spanning-tree mst config 命令查看这些信息。

一个域可以有一个或多个有相同 MST 配置的成员；每一个成员必须有处理 RSTP BPDU 的能力。在一个网络中没有限制 MST 域的数量，但是每个域最多支持 16 个实例。你一次只能分配一个 VLAN 到一个生成树实例中。

6.1.3 IST, CIST, 和 CST

内部生成树（IST），运行在 MST 域内的生成树。

在每一个 MST 域，MSTP 维护多个生成实例。实例 0 是一个域的一个特殊的实例，被称之为 IST。所有其他 MST 实例是数字 1 到 15。

这个 IST 仅仅是一个接收和发送 BPDU 的生成树实例；所有其他生成树实例信息被压缩

在 MSTI BPDU 中。因为 MSTI BPDU 携带所有实例的信息，需要被一个支持多生成树实例的交换机处理 BPDU 的数量意味着要简化。

所有在相同域的 MST 实例共享相同的协议 timer，但是每个 MST 实例有它自己的拓扑参数，例如一个根交换机 ID，根路径花消等等。缺省情况，所有的 VLAN 被分配到 IST。

公用的和内部的生成树（CIST），是每一个 MST 域里的所有 IST，和（连接 MST 域和单个生成树的）公用生成树的集合。

在一个域内计算的生成树看起来像是包含所有交换机域的 CST 的一个子树。CIST 被支持 802.1W 和 802.1D 协议的交换机之间的生成树计算运行的结果形成的。在 MST 域的 CIST 和在域外的 CST 相同。

公共生成树（CST），运行在 MST 域间的生成树。

6.1.4 域内操作

IST 连接一个域内所有 MSTP 交换机。当 IST 收敛时，IST 的根成为 IST master，它是域内最低 bridge ID 和到 CST 根的路径开销的交换机。如果在网络中只有一个域，IST master 也是 CST 根。如果 CST 根在域外，在域的边界（boundary）的一个 MSTP 交换机被选为 IST master。

当一个 MSTP 交换机初始化时，它发送 BPDU 要求它自己作为 CST 根和 IST master，到 CST 根和 IST master 的路径花销设置为 0。交换机也初始化所有的 MST 实例并且要求成为他们的根。如果交换机收到的 MST 根信息比当前端口存储的信息优先（低 bridge ID，低路径花消等等），它放弃它的成为 IST master 的要求。

在初始化中，一个域可能有许多亚域，每一个带有它自己的 IST master。当交换机收到一个更优先的 IST 信息时，它离开它旧的亚域加入到新的可能包含真是 IST master 的亚域。因此所有的亚域都收缩，包含真正的 IST master 的亚域除外。

为了正确的操作，所有的 MST 域内的交换机必须承认相同的 IST master。所以，任意两个域内的交换机同步他们的一个 MST 实例的端口的角色,只是如果它们收敛到一个公用 IST master。

6.1.5 域间操作

如果有多个域或早期 802.1D 交换机在网络中，MSTP 建立和维护 CST，它包含所有网

络中的 MST 域和所有早期 STP 交换机。MST 实例联合在域边界（boundary）的 IST 成为 CST.

IST连接所有 MSTP 域内的交换机并且看起来像 CST（包围所有交换机域）的一个子树，子树的根成为 IST master。MST 域看起来像一个虚拟的交换机邻接到 STP 交换机和 MST 域。

只不过 CST 实例发送和接收 BPDU，和 MST 实例增加它们的生成树信息到 BPDU 相互影响邻居交换机和计算最后的生成树拓扑。因为这个，涉及到 BPDU 传送 （比如：hello time, forward time, max-age,和 max-hops）的生成树参数被配置仅仅在 CST 实例但是不影响所有 MST 实例。涉及到生成树拓扑的参数（比如：switch priority, port VLAN cost , port VLAN priority）可以被配置在 CST 实例和 MST 实例。

MSTP 交换机使用版本 3 的 RSTP BPDU 或 802.1D 的 BPDU 和 802.1D 的交换机通信。MSTP 交换机使用 MSTP BPDU 和 MSTP 交换机通信。

6.1.6 跳的计数

在配置计算生成树拓扑的 BPDU 中 IST 和 MST 实例不使用 message-age 和 maximum-age 信息。替代为，使用到根的路径花消和相当于 IP TTL 的 hop-count 机制。

你可以配置那个域的最大跳数并应用到那个域 IST 和所有的 MST 实例。跳数计算实现和 message-age 结果相同（在引发一个重新配置后决定）。实例根交换机总是发送一个 cost 为 0, hop-count 为最大值的 BPDU（or-M-record)。当一个交换机收到 BPDU 时，它把剩余的跳数减 1， 并且在它产生的 BPDU 里面传播这个剩余的跳数。当计数到达 0，交换机丢弃 BPDU 并且 age 这个端口的信息。

在一个域里面，在 RSTP BPDU 部分里面的 Message-age 和 maximum-age 信息保留一致，相同的值被在边界（boundary）的域的指定端口传播。

6.1.7 边界端口

边界端口（boundary）是一个连接 MST 域到一个单独运行 RSTP 的生成树域，或者一个单独 801.1D 的生成树域，或者其他不同配置的 MST 域。一个边界端口也连接到一个 LAN，这个 LAN 的指定交换机要么是一个单独生成树交换机要么是一个带有不同的 MST 域配置的交换机。

在边界端口，MST 端口角色不重要，它们的状态被强制和 IST 端口状态相同（当 IST 端口是 forwarding 时，在边界的 MST 端口是 forwarding)。一个在边界的 IST端口可以有除备份端口以外的任何角色。

在一个共享边界连接，MST 端口在转换到 learning 状态前，在 blocking 状态等待 forward-delay time 到期。MST端口在转换到 forwarding 前，等待又一个 forward-delay time 到期。

如果边界端口是一个点到点连接并且是 IST 根端口，IST 端口一转换到 forwarding 状态 MST 端口就转换到 forwarding 状态。

如果一个边界端口在实例中转换到 forwarding 状态，它在所有的实例中都是 forwarding，一个拓扑改变是触发的。如果一个带有 IST 根或指定端口角色的边界端口接收到一个拓扑改变通告，MSTP 交换机在活跃的那个端口上的 IST 实例和所有 MST实例触发一个拓扑改变。

6.1.8 MSTP 和 802.1d STP 的互用性

一个运行 MSTP 的交换机支持一个内置的协议迁移机制，这个机制使他能够和 802.1D 协调使用。如果交换机从一个端口收到一个 802.1D 配置的 BPDU，它就在那个端口发送 802.1D BPDU。当一个域的边界端口收到一个802.1D BPDU 一个不同域的 MSTP BPDU 或 RSTP BPDU 时，MSTP 交换机可以侦察到。

然而，如果交换机不再接收 802.1D BPDU 它不会自动恢复到MSTP 模式，因为它不能决定对方的交换机是否已经从连接删除，除非对方的交换机是指定交换机。同样，当连接到这个交换机的交换机已经加入到这个域时，交换机可能继续分配一个边界端口角色到一个端口。重新启动协议的迁移处理（强制和邻居交换机协商)。

如果所有在连接的对方的交换机是 RSTP 交换机，它们可以处理 MSTP BPDU 和处理 RSTP BPDU。因此，MSTP 交换机在边界端口或者发送一个版本 0 配置和 TCN BPDU 或版本 3MSTP BPDU。一个连接到 LAN 的边界端口，他的指定交换机要么是一个单独生成树交换机或是一个不同 MST 配置的交换机。

6.1.9 端口角色

MSTP 采用 RSTP 的快速收敛算法。下面结合 RSTP 简单介绍 MSTP 端口角色和快速收敛。

RSTP 提供指定端口角色和决定活动拓扑的快速收敛。RSTP 基于 IEEE802.1D STP 之上，选择高优先级交换机作为根交换机。当 RSTP 指定一个端口角色到一个端口时：

Root port — 当交换机转发包到根交换机时提供最优路径花消。

Designated port — 连接指定交换机。当转发从 LAN 到根交换机数据包产生最低的路径花消。指定交换机通过它连接到 LAN 的端口叫指定端口。

Alternate port— 提供一个当前根端口的到根交换机的替换路径。

Backup port—扮演一个指定端口到生成树叶子的路径的备份。一个 Backup 端口存在仅仅当两个端口一起连接在一个点到点的环路或当一个交换机有两个或多个连接到一个共享 LAN 段。

Disable port—在生成树操作中没有端口角色。

Master port — 位于域根或到总根的最短路径上，它是连接域到总根的端口。

根端口或指定端口角色包含在活动拓扑。替换端口或备份端口角色不包含于活动拓扑。

在一个稳定拓扑和固定端口角色的整个网络，RSTP 确保每一个根端口和指定端口立即迁移到 forwarding 状态当所有的替换端口和备份端口总是在 discarding 状态时。端口状态控制 forwarding 和 learning 处理。

快速收敛

在下列情况下 RSTP 提供快速恢复：交换机故障，端口故障或 LAN 故障，它为边缘端口，新的根端口和连接到一个点到点的连接提供快速恢复：

Edge ports — 如果你配置一个端口作为边缘端口，边缘端口立即迁移为 forwarding 状态。你可以打开它为边界端口仅仅当这个端口连接到一个单独的终端或者确定不需要计算生成树的设备上。

Root ports — 如果 RSTP 选择一个新的根端口。它阻塞一个旧的根端口并且立即迁移新根端口到 forwarding 状态。

Point-to-point links — 如果你连接一个端口到其它端口通过一个点到点连接并且本地端口成为一个指定端口，它和其它端口经过 proposal-agreement 握手协商一个快速迁移确定一个快速收敛无回环（loop-free）拓扑。

拓扑改变

这部分描述 RSTP 和 802.1D 在处理 spanning-tree 拓扑改变的不同。

Detection — 不像802.1D在blocking和forwarding状态之间的任意迁移都会引起拓扑改变，仅仅从blocking迁移到forwarding状态导致RSTP 拓扑改变（只是为了增加连通性被考虑拓扑改变）。状态改变在一个边缘的端口（edge port）不会引起拓扑改变。当一个RSTP

交换机侦查到一个拓扑修改，它泛洪它学习到信息到所有的非边缘端口（nonedge ports）除了接收TC信息的端口外。

Notification — 不像802.1D，使用TCN BPDU，RSTP不使用它。然而，为了和802.1D的互用性，RSTP 交换机处理并产生TCP BPDU。

Acknowledgement — 当一个RSTP交换机在指定端口接收到一个来自802.1D交换机的TCN信息，它回应一个带有802.1D BPDU 并且设置TCA标志位。然而，如果TC-while timer(与802.1D 的topology-change timer相同)是活动的，在根端口连接到802.1D 交换机并收到一个带有TCA的配置BPDU，TC-while timer重起（reset）。这个行为只是被要求支持802.1D 交换机。RSTP BPDU从来都没有TCA标志位。

Propagation — 当RSTP交换机通过一个指定端口或根端口从其它交换机接收到一个TC信息，它传播到所有非边缘端口，指定端口和根端口（除接收端口以外的）。交换机所有这样的端口启动TC-while timer并且泛洪他们学习的信息。

Protocol migration — 为了向后兼容802.1D交换机，RSTP基于每一个端口选择性发送802.1D 配置BPDU和TCN BPDU。

当一个已经初始化，migrate-delay timer启动（指定最小值在RSTP BPDU被发送期间），RSTP BPDU被发送。当这个timer 是活动的，交换机处理所有的从端口接收的BPDU并且忽略协议类型。

在端口的migration-delay timer已经中止以后，如果交换机收到一个802.1D BPDU，它假设它连接到一个802.1D 交换机并且启动使用802.1D协议BPDU。然而，如果RSTP 交换机正在一个端口使用802.1D BPDU，在timer中止后接收到一个RSTP BPDU，那个端口它重新启动timer并且开始使用RSTP BPDU。

6.1.10 802.1D 生成树简介

生成树协议基于以下几点：

1）有一个唯一的组地址（01-80-C2-00-00-00）标识一个特定 LAN 上的所有的交换机。这个组地址能被所有的交换机识别；

2）每个交换机有一个唯一的标识（Brideg Identifier）；

3）每个交换机的端口有一个唯一的端口标识（Port Identifier）。对生成树的配置进行管理还需要：对每个交换机调协一个相对的优先级；对每个交换机的每个端口调协一个相对的优先级；对每个端口调协一个路径花费。

具有最高优先级的交换机被称为根（root）交换机。每个交换机端口都有一个根路径花

费，根路径花费是该交换机到根交换机所经过的各个网段的路径花费的总和。一个交换机中根路径花费的值为最低的端口称为根端口，若有多个端口具有相同的根路径花费，则具有最高优先级的端口为根端口。

在每个 LAN 中都有一个交换机被称为指定（designated）交换机，它属于该 LAN 中根路径花费最少的交换机。把 LAN 和指定交换机连接起来的端口就是 LAN 的指定端口（designated port）。如果指定交换机中有两个以上的端口连在这个 LAN 上，则具有最高优先级的端口被选为指定端口。

形成一个生成树所必需决定的要素：

1）决定根交换机

a、最开始所有的交换机都认为自己是根交换机；

b、交换机向与之相连的 LAN 广播发送配置 BPDU，其 root_id 与 bridge_id 的值相同；

c、当交换机收到另一个交换机发来的配置 BPDU 后，若发现收到的配置 BPDU 中 root_id 字段的值大于该交换机中 root_id 参数的值，则丢弃该帧，否则更新该交换机的 root_id、根路径花费 root_path_cost 等参数的值，该交换机将以新值继续广播发送配置 BPDU。

2）决定根端口

一个交换机中根路径花费的值为最低的端口称为根端口。

若有多个端口具有相同的最低根路径花费，则具有最高优先级的端口为根端口。若有两个或多个端口具有相同的最低根路径花费和最高优先级,则端口号最小的端口为默认的根端口。

3）认定 LAN 的指定交换机

a、开始时，所有的交换机都认为自己是 LAN 的指定交换机。

b、当交换机接收到具有更低根路径花费的（同一个 LAN 中）其他交换机发来的 BPDU，该交换机就不再宣称自己是指定交换机。如果在一个 LAN 中，有两个或多个交换机具有同样的根路径花费，具有最高优先级的交换机被选为指定交换机。

c、如果指定交换机在某个时刻收一了 LAN 上其他交换机因竞争指定交换机而发来的配置 BPDU，该指定交换机将发送一个回应的配置 BPDU，以重新确定指定交换机。

4）决定指定端口

LAN 的指定交换机中与该 LAN 相连的端口为指定端口。若指定交换机有两个或多个端口与该 LAN 相连，那么具有最低标识的端口为指定端口。

除了根端口和指定端口外，其他端口都将置为阻塞状态。这样，在决定了根交换机、交

换机的根端口、以及每个 LAN 的指定交换机和指定端口后，一个生成树的拓扑结构也就决定了。

6.2 MSTP 配置

6.2.1 缺省配置

命令参数	缺省值
spanning-tree mst enable(启动 mstp)	关闭
Spanning-tree mst priority(交换机 cist 优先级)	32768
spanning-tree mst hello-time(交换机 cist hello-time)	2 秒
spanning-tree mst forward-time(交换机 cist forward-time)	15 秒
spanning-tree mst max-age(交换机 cist max-age)	20 秒
spanning-tree mst max-hops(交换机 cist max-hops)	20 秒
instance 1 priority (实例优先级)	32768
spanning-tree mst instance 1 priority(端口实例 priority)	128
spanning-tree mst instance 1 path-cost(端口实例 path-cost)	20000000
spanning-tree mst priority （端口 cist priority）	128
spanning-tree mst path-cost（端口 cist path-cost）	20000000

6.2.2 一般配置

启动MSTP

系统在启动时缺省配置MSTP 是关闭的。

启动MSTP 的配置过程是：

Switch#configure terminal

Switch(config)#spanning-tree mst enable

关闭MSTP**的命令是：**

Switch#configure terminal

Switch(config)#no spanning-tree mst

配置max-age

配置max-age**是对所有的实例的配置，**max-age**是交换机在触发一个重新配置前，等待接收生成树配置信息的秒数。**

缺省配置是20**秒，配置范围是**6**到**40**秒。**

配置过程：

Switch#configure terminal

Switch(config)#spanning-tree mst max -age <seconds>

配置max-hops

max-hops**是在一个域中在**BPDU**被丢弃前指定的跳数。**

缺省值是20**，配置范围是**1**到**40**。**

配置过程：

Switch#configure terminal

Switch(config)#spanning-tree mst max -hops <hop-count>

配置forward-time

配置forward-time**是对所有的实例。**forward-time**是端口从**discarding**到**learning**以及**learning**到** forwarding**等待的秒数。**

缺省配置是15**秒，配置范围是**4**到**30**秒。根据生成数协议**forward-time **必须满足下列条件：**2*(forward-time – 1) >= max -age。

配置过程：

Switch#configure terminal

Switch(config)#spanning-tree mst forward-time <seconds>

配置hello-time

配置hello-time**是对所有实例的配置。**hello-time**是根交换机产生配置信息的间隔时间。**

缺省配置时间是2**秒，配置范围是**1**到**10**秒。根据生成数协议**hello-time **必须满足下列条件：**2*(hello -time + 1) =< max -age。

配置过程：

Switch#configure terminal

```
Switch(config)# spanning-tree mst hello-time <seconds>
```

配置CIST bridge**的优先级（**priority**）**

缺省配置32768**、配置范围**<0-61440>**；**CIST**优先级的值只能是**4096**的倍数。**

配置过程：

```
Switch#configure terminal
Switch(config)#spanning-tree mst priority <priority>
```

配置和CISCO**兼容**

联想天工网络交换机采用基于802.1s**的**MSTP**协议，每个**MSTI**消息的长度是**16**个字节；而**CISCO**交换机的**BPDU**每个**MSTI**消息的长度是**26**个字节。为了和**CISCO**交换机互用，配置联想天工网络的交换机时要启动和**CISCO**兼容的开关。**

在启动和CISCO**兼容配置的情况下，在判断是否为相同的域时，只要域名和修订版本号相同就认为是相同的域。**

缺省系统不启动这个功能。

打开和CISCO**兼容：**

```
Switch#configure terminal
Switch(config)#spanning-tree mst cisco-interoperability enable
```

关闭和CISCO**兼容：**

```
Switch#configure terminal
Switch(config)#spanning-tree mst cisco-interoperability disable
```

复位协议检查任务

为了和802.1D STP**协议的兼容，系统可以自动侦察对方系统运行的协议。根据对方运行的协议来决定这个端口运行的协议。**

有些情况下要复位协议。例如系统经过协商一个端口运行STP**协议，一段时间后对方的运行**STP**协议的设备已替换为一台主机。这时我需要配置这个端口为**fast port**，但是该端口已经运行了**stp**协议，而且协议协商的任务已经停止；这时需要复位这个协议协商的任务让它重新协商它和主机之间的协议。**

复位整个设备的协议侦察任务：

```
Switch#clear spanning-tree detected protocols
```

复位某个端口的协议侦察任务：

```
Switch#clear spanning-tree detected protocols interface <if-name>
```

6.2.3 域配置

两个或者多个设备在相同的域，他们必须有相同的 VLAN **实例映射关系，相同的修改版本号和相同的域名。**

一个域有一个或多个有相同 MST **配置的成员，每个成员都可以处理** RSTP BPDUS **能力。在一个网络中不限制成员数量，但是每个域最多能够支持** 16 **个实例。**

关于实例的配置在'实例配置'里面说明，这里只介绍域名配置和修订版本号配置。

配置域名：

```
Switch#configure terminal
Switch(config)#spanning-tree mst configuration
Switch(config-mst)#region <region-name>
```

配置修订版本号：

```
Switch#configure terminal
Switch(config)#spanning-tree mst configuration
Switch(config-mst)# revision <revision-num>
```

6.2.4 实例配置

系统支持 16 **个实例，实例** ID **号的范围是** 0-15**。一个** VLAN **一次只能分配到一个生成树实例。**

缺省情况只存在一个实例0**，所有的**VLAN**都属于这个实例。**

配置一个实例的过程：

```
Switch#configure terminal
Switch(config)#spanning-tree mst configuration
Switch(config-mst)#instance <instance-id> vlan <vlan-id>
```

配置MSTI bridge**的优先级（**priority**）**

缺省配置 32768**、配置范围**<0-61440>**；**MSTI **优先级的值只能是** 4096 **的倍数。**

配置过程：

```
Switch#configure terminal
```

Switch(config)#spanning-tree mst configuration

Switch(config-mst)#instance <instance-id> priority <priority>

6.2.5 端口配置

下面介绍MSTP **相关的端口配置信息。这里只介绍简单配置部分，**port fast**和**root guard **在后面单独介绍。**

配置一个端口加入到一个实例的过程：

Switch#configure terminal

Switch(config)#interface <if-name>

Switch(config)#spanning-tree mst instance <instance-id>

配置CIST **端口的优先级（**priority**）**

缺省配置128**，配置范围是**<0-240>,CIST**端口的优先级的值只能是**16**的倍数。**

配置过程：

Switch#configure terminal

Switch(config)#interface <if-name>

Switch(config)#spanning-tree mst priority <priority>

配置MSTI**端口的优先级（**priority**）**

缺省配置128**，配置范围是**<0-240>,MSTI**端口的优先级的值只能是**16**的倍数。**

配置过程：

Switch#configure terminal

Switch(config)#interface <if-name>

Switch(config)#spanning-tree mst instance <instance-id> priority <priority>

配置CIST**端口的路径花消（**path-cost**）**

缺省配置 20000000**，配置范围是** 1-200000000**。下面是带宽和路径花消映射表：**

带宽(bps)	路径花消
100,000(100K)	200000000
1,000,000(1M)	20000000
10,000,000(10M)	2000000

100,000,000(100M)	200000
1,000,000,000(1G)	20000
10,000,000,000(10G)	2000
100,000,000,000(100G) 1,000,000,000,000(1T)	200 20
>1000000000000	2

配置过程

Switch#configure terminal

Switch(config)#interface <if-name>

Switch(config)#spanning-tree mst path <path-cost>

配置MSTI**端口的路径花消（**path-cost**）**

缺省配置20000000**，配置范围是**1-200000000**。带宽和路径花消和上面的表一样。**

配置过程

Switch#configure terminal

Switch(config)#interface <if-name>

Switch(config)#spanning-tree mst instance <instance-id> path-cost <path-cost>

配置发送协议包的版本号

缺省配置发送MSTP**协议包，配置范围是**0-3,**映射关系是**0-stp,2-rstp,3-mstp**。**

配置过程：

Switch#configure terminal

Switch(config)#interface <if-name>

Switch(config)# spanning-tree mst force-version <version-id>

配置连接类型

如果一个端口通过点到点的方式连接到其它的端口，并且本地端口成为一个designated port**（指定端口）**,RSTP**通过**proposal-agreement(**提议**-**协定**)**过程协商一个快速迁移它所连接的端口成为根端口来确定一个无环的拓扑。**

下面简单介绍proposal-agreement**的协商过程。**

当交换机在它的一个端口收到一个提议信息并且那个端口被选择为新的根端口，RSTP **强迫所有其它端口同步新根端口信息。**

如果其它所有的端口被从根端口接收的更优的（superior**）根信息同步，那么交换机被同步。**

当RSTP 强制它同步新的根信息时，如果一个指定端口是在forwarding状态，而且没有被配置为一个边缘端口，它迁移到blocking状态。通常，当RSTP 强制一个端口同步新的根消息并且端口不能满足上面的条件时，端口状态设置为blocking。

当确保所有的端口被同步，交换机发送一个agreement 信息到根端口相应的指定端口。当交换机连接到一个点到点连接在agreement他们的端口角色，RSTP 立即迁移端口状态为forwarding。

如果是共享连接，则要经过802.1D 的计算过程来确定端口的状态。

缺省情况端口连接类型是点到点的连接。

配置端口的连接类型是点到点的连接：

```
Switch#configure terminal
Switch(config)#interface <if-name>
Switch(config)#spanning-tree mst link-type point -to-point
```

配置端口的连接类型是共享连接：

```
Switch#configure terminal
Switch(config)#interface <if-name>
Switch(config-ge2/1)#spanning-tree mst link -type shared
```

6.2.6 PORTFAST 相关配置

1) Port Fast

Port Fast 立即转移一个 access 或 trunk 端口从 blocking 状态到 forwarding 状态，绕过 listening 和 learning 状态。你可以用 Port Fast 在连接一个单独工作站和服务器，可以允许这些设备立即连接到网络，不需要等待 spanning tree 收敛。

配置一个端口为 fast port：

```
Switch#configure terminal
Switch(config)#interface <if-name>
Switch(config)#spanning-tree mst portfast
```

2) BPDU Filtering

BPDU filtering可以基于交换机全局打开或者基于每个端口打开，但是他们的特点是不

一样的。

在全局层，你可以用spanning-tree mst portfast bpdu-filter命令启动在portfast bpdu-filter default状态的端口的BPDU filtering功能。

在端口层，你可以用 spanning-tree mst portfast bpdu-filter enable 在任意端口打开 BPDU filter。

这个功能防止 port fast 端口接收或发送 BPDU。

配置 BPDU Filtering

在全局配置模式下：

Switch#configure terminal

Switch(config)# spanning-tree mst portfast bpdu-filter

在接口配置模式下：

Switch#configure terminal

Switch#interfac e <if-name>

Switch(config)#spanning-tree mst portfast bpdu-filter enable

3) BPDU Guard

BPDU 保护特性可以在交换机全局打开或是基于每个端口被打开，但是他们的特点是不一样的。

在全局层,你可以用 spanning-tree mst portfast bpdu-guard 打开在 portfast bpdu-gurad default 状态的端口的 BPDU guard 功能。

在端口层，你可以在任何端口打开 BPDU guard。

当配置了 BPDU gurad 的端口收到 BPDU 时，spanning tree 会 shutdown 这个的端口。在一个有效的配置，Port Fast-enabled 的端口不接收 BPDU。在一个 Port Fast-enabled 的端口接收到一个 BPDU 表示一个无效的配置，例如是一个未授权设备的连接，BPDU guard 进入到一个 error-disabled 状态。

error-disabled 是当启动 BPDU gurad 的端口收到 BPDU 时,如果系统配置 error-disable 机制时会启动 error-disable timer。error-disable 会在系统配置的超时时间后重新启动这个端口。

在全局配置模式下：

Switch#configure terminal

Switch(config)# spanning-tree mst portfast bpdu-guard

在接口配置模式下：

Switch#configure terminal

```
Switch#interface <if-name>
Switch(config)#spanning-tree mst portfast bpdu-guard enable
```

error-disable 的配置

启动 error-disable 机制

```
Switch#configure terminal
Switch#spanning-tree mst errdisable-timeout enable
```

配置 error-disable 超时时间

```
Switch#configure terminal
Switch# spanning-tree mst errdisable-timeout interval <seconds>
```

6.2.7 Root Guard 相关配置

一个 SP 的二层网络可以包含许多连接到不属于他们自己的交换机。在这样一个拓扑，生成树可以重新配置它自己并且选择一个客户交换机作为根交换机。你可以通过配置 root guard 在 SP 交换机的连接到在客户网络的交换机的端口避免这种情况。如果生成树计算导致在客户网络的端口被选为 root port，root guard 就配置端口为 root-inconsistent(blocked) 状态防止客户交换机成为根交换机或存在到根的路径。

如果一个 SP 网络外面的交换机成为根交换机，端口是 blocked（root-inconsistent stat）并且生成树选择一个新的根交换机。客户的交换机不会成为根交换机并且不存在到根的路径。

如果交换机在 MST 模式操作，root guard 强制端口成为指定端口。如果一个边界端口因为 root guard 在 IST 实例是 blocked 状态，这端口在所有的 MST 实例是 block 的。一个边界端口是连接到一个 LAN 的端口，指定交换机要么是一个 802.1D 交换机或一个不同 MST 域配置的交换机。

在一个端口被打开 Root guard 应用到所有的这个端口所属的 VLAN。VLAN 可以被聚合并映射到一个 MST 实例。

配置过程

```
Switch#configure terminal
Switch(config)#interface <if-name>
Switch(config)#spanning-tree mst guard root
```

6.3 MSTP 配置示例

(1)配置

三台交换机连接成一个环状，需要打开每一台交换机的生成树协议，避免环路的发生。分别在每一台交换机上执行配置。

交换机1**的配置：**

Switch>en

Switch#configure terminal

Switch(config)#spanning mst enable

交换机2**的配置：**

Switch>en

Switch#configure terminal

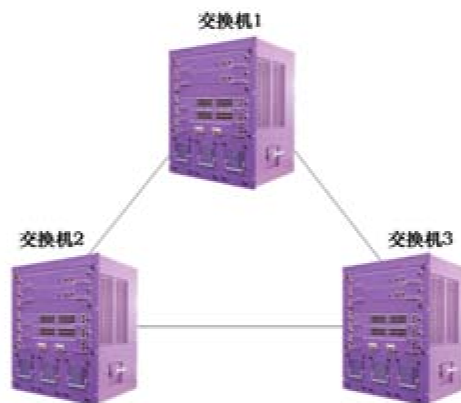
Switch(config)#spanning mst enable

交换机3**的配置：**

Switch>en

Switch#configure terminal

Switch(config)#spanning mst enable



(2)排错：

查看哪一个交换机被选为根网桥：

执行show spanning-tree mst**，观察**CISTRoot**的值为三台交换中**MAC**地址最小的一个，**

即根选举结果正确。

Switch#show spanning -tree mst

查看生成树中交换机的端口状态：

执行show spanning-tree mst interface ge1/1**这条指令，观察**PORT ge1/1**在实例**0**中的** State **值**

Switch#show spanning -tree mst interface ge1/1

第7章 配置IGMP SNOOPING

在城域网/Internet中，采用单播方式将相同的数据包发送给网络中的多个而不是全部接收者时，由于需要复制分组给每一个接收端点，随着接收者数量的增多，需要发出的包数也会线性增加，这使得主机、交换路由设备及网络带宽资源总体负担加重，效率受到极大影响。随着多点电视会议、视屏点播、群组通信应用等需求的增长，为提高资源利用率，组播方式日益成为多点通信中普遍采用的传输方式。

iSpirit 12800系列交换机实现了IGMP SNOOPING功能，为组播应用服务。IGMP SNOOPING监听网络上的IGMP包，实现IP组播MAC地址的动态学习。

本章对IGMP SNOOPING的概念和配置进行描述，主要包括以下内容：

- IGMP SNOOPING 介绍
- IGMP SNOOPING 配置
- IGMP SNOOPING 配置示例

7.1 IGMP SNOOPING 介绍

传统的网络在一个子网内组播数据包当作广播处理，这样容易使网络流量大，造成网络拥塞。当交换机上实现了IGMP SNOOPING后，IGMP SNOOPING可以动态学习IP组播MAC地址，维护IP组播MAC地址的输出端口列表，使组播数据流只往输出端口发送，这样可以减少网络的流量。

本节主要包括以下内容：

- IGMP SNOOPING 处理过程
- 二层动态组播
- 加入一个组
- 离开一个组

7.1.1 IGMP SNOOPING 处理过程

IGMP SNOOPING是一个二层的网络协议，监听经过交换机的IGMP 协议包，根据这些IGMP协议包的接收端口，VLAN ID和组播地址来维护一个组播组，然后转发这些IGMP 协议包。只有加入了组播组的端口才可以接收组播数据流；这样就减少了网络的流量，节省了网络带宽。

组播组包括了组播组地址，成员端口，VLAN ID，Age时间。

IGMP SNOOPING组播组的形成是一个学习的过程。当交换机的某一个端口收到IGMP REPORT包时，IGMP SNOOPING会产生一个新的组播组，接收IGMP REPORT包的端口就被加入这个组播组。在交换机收到一个IGMP QUERY包时，如果这个组播组已经存在交换机中，那么这个收到IGMP QUERY的端口也加入到这个组播组中，否则只是转发IGMP QUERY包。IGMP SNOOPING还支持IGMP V2的Leave机制；如果IGMP SNOOPING配置了fast-leave 为ENABLE，在收到IGMP V2的leave包时它接收端口可以立刻离开组播组；如果配置了fast-leave离开等待时间(fast-leave-timeout)，那么组播组在等待这个时间到期后再离开组播组。

IGMP SNOOPING有两种更新机制。一种是上面介绍的leave机制。大多数情况下IGMP SNOOPING是通过age time来删除过期的组播组的。当组播组加入IGMP SNOOPING时记录了加入的时间，当组播组在交换机中存留的时间超过了一个配置的age time时，交换机会删除这个组播组。

当一个端口收到Leave协议包时，这个端口会立即从它所属的组播组中删除，这种情况可能会影响网络数据流的连续性；因为这端口下面可能连接着一个HUB或没有IGMP SNOOPING功能的网络设备，这个设备下连接了很多的接收组播数据流设备。一个设备发送Leave，可能会影响其他设备也接收不到组播数据流。Fast-leave-timeout机制可以防止这种情况的发生，通过Fast-leave-timeout配置一个离开等待的时间，端口收到leave包后等待Fast-leave-timeout长的时间再从它所属的组播组中删除，可能保障网络组播流的连续性。

7.1.2 二层动态组播

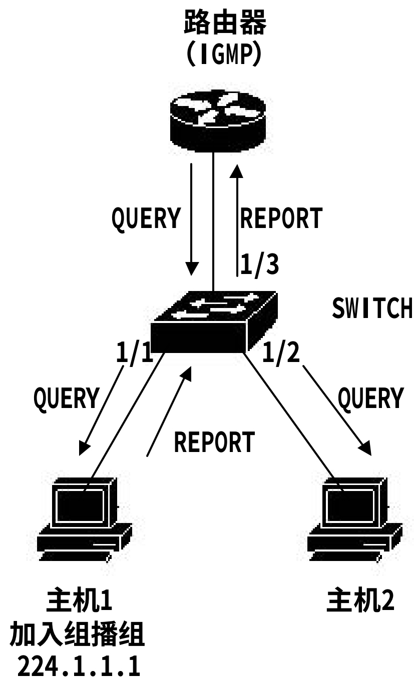
二层硬件组播转发表中的组播MAC地址条目可以通过IGMP SNOOPING动态学习得到。通过IGMP SNOOPING动态学习到的是IP组播MAC地址。

当交换机关闭IGMP SNOOPING时，二层硬件组播转发表处于未注册转发模式，组播MAC地址不能动态学习到，二层硬件组播转发表中没有条目，所有的二层组播数据流当作广播处理。

当网络具备组播环境时，为了有效控制网络的组播流量，交换机可以打开IGMP SNOOPING，此时二层硬件组播转发表处于注册转发模式，交换机可以通过监听网络上的IGMP协议包学习到组播MAC地址，与二层硬件组播转发表中的条目匹配的二层组播流才能够转发。

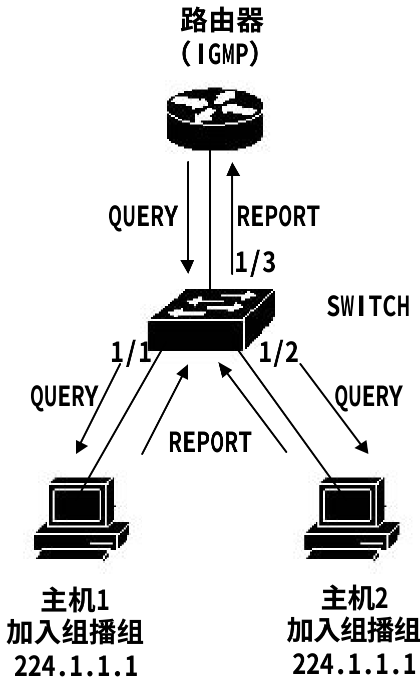
7.1.3 加入一个组

当一个主机想加入一个组播组时，主机会发一个IGMP REPORT包，在此包中指定主机要加入的组播组。当交换机收到一个IGMP QUERY包时，交换机会把该包转发给同一个VLAN的所有其它端口，当端口下的想加入组播组的主机收到IGMP QUERY包后会回送一个IGMP REPORT包。当交换机收到一个IGMP REPORT包后，会建立一个二层组播条目，收到IGMP QUERY包的端口和IGMP REPORT包的端口会加入到该二层组播条目，成为它的输出端口。



如上图所有的设备在一个子网内，假设该子网的VLAN**是**2**。路由器运行**IGMPv2**协议，定时发送**IGMP QUERY**包。主机**1**想加入组播组**224.1.1.1**。交换机从**1/3**端口收到**IGMP QUERY**包后会记录此端口并把该包转发给端口**1/1**和**1/2**。主机**1**收到**IGMP QUERY**包后回送一个**IGMP REPORT**包，主机**2**因为不想加入组播组，不发**IGMP REPORT**包。交换机从端口**1/1**收到**IGMP REPORT**包后会把该包从查询端口**1/3**转发出去并且创建一个二层组播条目（假定该条目不存在），该二层组播条目包括以下几项：**

二层组播地址	VLAN ID	**输出端口列表**
01:00:5e:01:01:01	2	1/1，1/3



如上图的条件与图1一样，主机1已经加入了组播组224.1.1.1，现在主机2想加入组播组224.1.1.1。当主机2收到IGMP QUERY包后回送一个IGMP REPORT包，交换机从端口1/2收到IGMP REPORT后会把该包从查询端口1/3转发出去并且会包端口1/2加入到二层组播条目中，该二层组播条目变为：

二层组播地址	VLAN ID	输出端口列表
01:00:5e:01:01:01	2	1/1，1/2，1/3

7.1.4 离开一个组

为了能够组成一个稳定的组播环境，运行IGMP的设备（如路由器）会每隔一定的时间发送一个IGMP QUERY包给所有的主机。已经加入组播组或想加入组播组的主机收到该IGMP QUERY后会回送一个IGMP REPORT。

如果主机想离开一个组播组，可以有两种方式：主动离开和被动离开。主动离开就是主机发送一个IGMP LEAVE包给路由器，被动离开就是当主机收到路由器发来的IGMP QUERY后不回送IGMP REPORT。

与主机离开组播组的方式对应，在交换机上端口脱离二层组播条目的方式也有两种：超

时离开和收到IGMP LEAVE包离开。

当交换机超过一定的时间没有从一个端口收到一个组播组的IGMP REPORT包时，该端口要从对应的二层组播条目中清除，如果该二层组播条目没有了端口，则删除二层组播条目。

当交换机的fast-leave配置为ENABLE 时，如果某个端口收到一个组播组的IGMP LEAVE 包时，该端口从对应的二层组播条目中清除，如果该二层组播条目没有了端口，则删除此二层组播条目。

Fast-leave 一般应用在一个端口下接一个主机的情况；如果一个端口下面多于一个主机，可以配置fast-leave-timeout等待时间，这样可以保证网络中组播流的连续性和可靠性。

7.2 IGMP SNOOPING 配置

7.2.1 IGMP SNOOPING 缺省配置

IGMP SNOOPING缺省是关闭的，二层硬件组播转发表处于未注册转发模式。

Fast-leave缺省是关闭的。

Fast-leave-timeout 时间为300秒。

组播组REPORT 端口的age时间缺省为300秒。

组播组QUERY端口的age时间缺省为400秒。

7.2.2 打开和关闭 IGMP SNOOPING

打开IGMP SNOOPING协议可以全局打开也可以单独打开部分VLAN；只有全局打开IGMP SNOOPING才能打开或关闭某个VLAN的IGMP SNOOPING。

打开全局IGMP SNOOPING

Switch#configure terminal

Switch(config)#ip igmp snooping

打开一个VLAN的IGMP SNOOPING

Switch#configure terminal

Switch(config)#ip igmp snooping vlan <vlan-id>

关闭全局IGMP SNOOPING

Switch#configure terminal

Switch(config)#no ip igmp snooping

关闭一个VLAN**的**IGMP SNOOPING

Switch#configure terminal

Switch(config)#no ip igmp snooping vlan <vl an-id>

7.2.3 配置生存时间

配置组播组的生存时间

Switch#configure terminal

Switch(config)#ip igmp snooping group-membership-timeout <interval> vlan <vlan-id>

Interval**的单位是毫秒。**

配置查询组的生存时间

Switch#configure terminal

Switch(config)#ip igmp snooping query-membership-timeout <interval> vlan <vlan-id>

Interval**的单位是毫秒。**

7.2.4 配置 fast-leave

启动一个VLAN**的**fast-leave

Switch#configure terminal

Switch(config)#ip igmp snooping fast-leave vlan <vlan-id>

关闭fast-leave

Switch#configure terminal

Switch(config)#no ip igmp snooping fast-leave vlan <vlan-id>

配置fast-leave**等待时间**

Switch#configure terminal

Switch(config)# ip igmp snooping fast-leave-timeout <interval> vlan <vlan-id>

恢复缺省fast-leave**等待时间**

Switch#configure terminal

Switch(config)#no ip igmp snooping fast-leave-timeout vlan <vlan-id>

7.2.5 配置 MROUTER

配置静态的查询端口

Switch#configure terminal

Switch(config)#ip igmp snooping mrouter <interface-name> vlan [vlan-id]

如果用户没有输入VLAN**，那么系统会把这查询端口配置为这个端口所属的所有**VLAN**的**MROUTER。

7.2.6 显示信息

显示IGMP SNOOPING**配置信息**

Switch#show ip igmp snooping

显示一个VLAN**的配置信息**

Switch#show ip igmp snooping vl an <vlan-id>

显示REPORT**组播组的老化信息**

Switch#show ip igmp snooping age-table group-membership

显示QUERY**的老化信息**

Switch#show ip igmp snooping age-table query-membership

显示组播组的转发信息

Switch#show ip igmp snooping forwarding-table

显示MROUTER**信息**

Switch#show ip igmp snooping mrouter

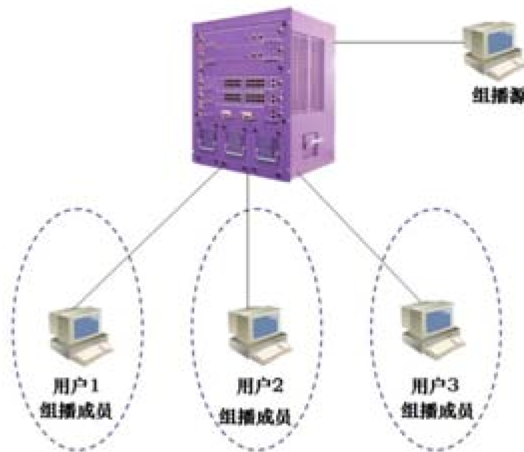
显示系统当前配置，包括IGMP SNOOPING**的配置**

Switch#show running-config

7.3 IGMP SNOOPING 配置示例

7.3.1 配置

在交换机上启用IGMP SNOOPING**功能，用户**1**、用户**2**、用户**3**可加入到特定的组播组中。**




```
Switch#config t
Switch(config)#ip igmp snooping
Switch(config)#vlan database
Switch(config-vlan)#vlan 200
Switch(config-vlan)#exit
Switch(config)#interface ge1/1
Switch(config-ge1/1)#switchport mode access
Switch(config-ge1/1)#switchport access vlan 200
```

```
Switch(config)#interface ge1/2
Switch(config-ge1/2)#switchport mode access
Switch(config-ge1/2)#switchport access vlan 200
Switch(config)#interface ge1/3
Switch(config-ge1/3)#switchport mode access
Switch(config-ge1/3)#switchport access vlan 200
Switch(config)#ip igmp snooping vlan 200
Switch(config)#ip igmp snooping group-membership-timeout 60000 vlan 200
```

第8章 配置ACL

在实际的网络中，网络的访问安全是管理员非常关注的问题。iSpirit 12800系列交换机支持ACL过滤提供网络的访问安全。通过配置ACL规则，交换机根据这些规则对输入的数据流过滤实现网络的访问安全。

本章介绍如何配置ACL，主要包括以下内容：

- ACL资源库介绍
- ACL过滤介绍
- ACL资源库配置
- ACL过滤配置
- ACL配置示例

8.1 ACL 资源库介绍

ACL(Access list control)资源库是多组访问规则的集合，ACL资源库没有控制数据转发的功能，只是一个具有冲突排序的规则集合。ACL资源库在被应用引用后，这些应用就根据ACL资源提供的规则来控制数据的转发。ACL可以应用于端口访问过滤，服务访问过滤和QOS等等。

ACL资源库有标准IP规则组（组号1～99，1300～1999），扩展IP规则组（组号100～199，2000～2699）；每一组规则内部自动进行冲突规则优先顺序排序。当用户配置一个ACL规则时，系统会根据排序规则把这条规则插入到相应的位置。

在应用时，当一个数据包通过一个端口的时候，交换机将每一条规则中的字段和数据包中相应的所有字段进行比较；当同时出现多个规则完全匹配时，最先完全匹配的一条规则生效；由这条匹配的规则来决定数据包是转发还是丢弃。所谓的完全匹配是，规则中的字段的值和数据包中相应字段的值完全相等。只有完全匹配ACL某一条规则，这规则才会作相应的deny或permit操作。

在iSpirit 12800系列交换机中，同一组内的规则是自动排序的。规则的自动排序相对比较复杂，在排序过程中范围大的规则排在后面，范围小的排在前面。范围的大小由规则的约束条件决定；规则的约束条件越少规则匹配的范围就越大，规则的约束条件越多规则匹配的范围就越小。规则的约束条件主要体现在地址的wildcard和一些非地址字段的个数两方面。Wildcard是bit串。IP地址是四字节，MAC地址是六字节。bits为'1'表示不需要匹配，bits为'0'表示要匹配。非地址字段是指协议类型，IP协议类型，协议端口，这些字段也隐藏了一个wildcard。他们的长度是相应字段的字节长度，因此相同的字段长度是统一的，只需计算字段的个数。Wildcard为'0'的bit越多约束条件就越多。

下面以端口访问过滤为例说明规则排序的必要性和自动排序的优点。假如用户需要拒绝源地址为192.168.0.0/16网段的地址转发，允许源地址为192.168.1.0/24网段的地址转发，可以配置以下两条规则：

access-list 1 permit 192.168.1.0 0.0.0.255 —规则1

access-list 1 deny 192.168.0.0 0.0.255.255 —规则2

后面简称规则1和规则2。

这两条规则是有冲突的；因为规则2的地址包含在规则1的地址中，而且一个是deny，一个是permit；根据ACL的过滤原理，不同的顺序有不同的结果。如果要实现上述要求，上面两条规则的顺序必须是：规则1排在前面，规则2排在后面。交换机自动实现了上述的排序功能，无论用户以怎样的先后顺序配置上述的规则，最后的顺序都是规则1排在规则2的

前面。当一个源地址为192.168.1.1地址的包上来转发时，首先比较第一条规则，再往后进行比较第二条规则，两条规则都匹配，前面的生效(转发)；如果源地址为192.168.0.1时,只有第一条匹配，那么就丢弃(不转发)。如果没有进行排序，用户可能会现先配置规则2，后配置规则1；规则1排在后面，规则2排在前面。

access-list 1 deny 192.168.0.0 0.0.255.255—规则2

access-list 1 permit 192.168.1.0 0.0.0.255—规则1

因为前面的规则2包含了后面的规则1，可能会导致的情况是：完全匹配规则1的数据包也完全匹配规则2，规则2每次都会生效；而不能达到应用的需求。

在交换机中，'0.0.255.255'是Wildcard bits，bits为'1'表示不需要匹配，bits为'0'表示要匹配。由此可以看出规则2的Wildcard bits为'0.0.255.255'，需要匹配两个字节(16个bits)；规则1中，的Wildcard bits为'0 0.0.0.255'，需要匹配三个字节(24个bits)；所以规则2的规则'范围'更大，因此排在前面。在扩展IP中，排序需要考虑更多的规则字段，如IP协议类型,通信端口等等。它们的排序规则是一样的，即配置限制越多规则的'范围'就越小，反之'范围'就越大。规则的排序在后台实现，用户命令只能按用户配置的先后顺序显示。

ACL支持的过滤字段包括了源IP，目的IP，IP协议类型(如：TCP，UDP，OSPF)，源端口（如161），目的端口。用户可以根据不同的需要，配置不同的规则来进行访问控制。

在交换机中，一组规则可以被多个应用所应用；如：一组规则被端口访问过滤和服务访问过滤同时引用或同时被两个端口的端口访问过滤所引用。

8.2 ACL 过滤介绍

ACL过滤是在交换机的输入端口处进行的，对输入到此端口的数据流进行规则匹配实现端口的过滤。ACL过滤都是交换机的线速进行处理的，不会影响数据流的转发效率。

当交换机的某端口没有配置ACL过滤时，所有通过该端口输入的数据流不会进行规则匹配，可以通过该端口进行转发。当交换机的某端口配置了ACL过滤时，所有通过该端口的输入数据流会进行规则匹配，匹配的规则的动作如果是permit，该数据流允许转发，如果是deny，该数据流不允许转发，丢弃。

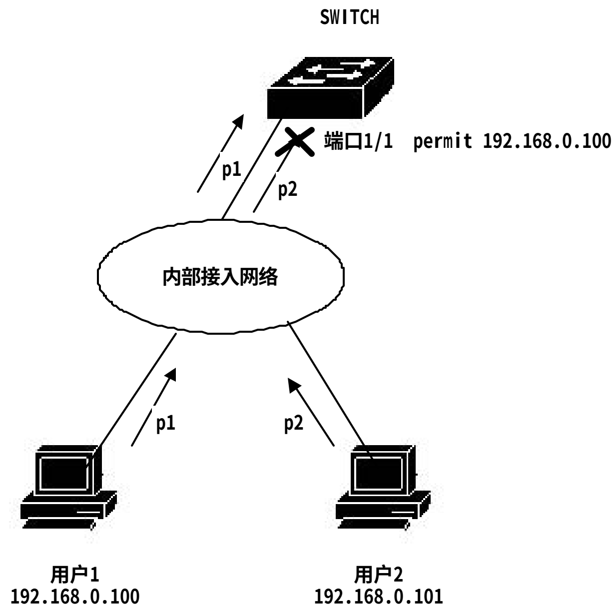
在配置端口的ACL过滤时，一个端口可以选择多个ACL规则组，选择后该组规则导入到端口的FFP中，如果该组规则中没有拒绝或允许所有IP协议包的规则，则写入FFP时会加一条拒绝所有IP协议的规则。当ACL资源库的规则变化后，写入FFP中的规则也会自动的变化。

例如一组规则中只有一条规则：access-list 1 permit 192.168.1.0 0.0.0.255，缺省会隐

藏一条拒绝所有IP协议包的规则，实际上会有两条规则导入到端口的FFP。在数据流过滤时，只有源地址从192.168.1.0到192.168.1.255的数据流可以通过该端口进行转发，所有其它的数据流被过滤掉。

例如一组规则中有两条规则：access-list 1 deny 192.168.1.0 0.0.0.255和access-list 1 permit any。此时有一条允许所有IP协议包的规则，这时不存在隐藏的规则，实际上会有两条规则导入到端口的FFP。在数据流过滤时，只有源地址从192.168.1.0到192.168. 1.255的数据流被过滤掉，所有其它的数据流被可以进行转发。

如下图是一个ACL过滤的例子。交换机的端口1/1选择一个ACL规则组1，该组规则中只有一条规则access-list 1 permit 192.168.0.100。在交换机的端口1/1下，有两个用户想从该端口接入网络，用户1的IP地址是192.168.0.100，用户2的IP地址是192.168.0.101。只有用户1可以通过交换机的端口1/1接入网络，用户2不能通过交换机的端口1/1接入网络。用户1发出来的数据流p1可以通过交换机的端口1/1转发，而用户2发出来的数据流p2则在交换机的端口1/1处丢弃。



多个端口做ACL过滤时可以选用同一个ACL规则组，使用相同的过滤规则。

不管是一组规则还是多组规则被一个端口引用，它们都会自动的进行排序，即使是两组规则之间的排序有交叉的情况。

当用户引用了一组规则后，如果这组规则发生变化，那么引用了这组规则的端口会自动响应用户的配置；不需要重新来配置这个端口的引用。

8.3 ACL 资源库配置

交换机缺省没有任何规则。

在交换机中的资源库支持三类ACL**规则：标准**IP**规则，扩展**IP**规则，扩展**MAC**规则。下面分三类规则来介绍**ACL**的配置。**

标准IP**规则：标准**IP**规则是通过源**IP**地址来控制数据包的转发。**

命令形式：access-list <groupId> {deny | permit} <source>

参数说明：

groupId **：访问控制列表组号，标准**IP ACL **支持从**1**到**199**组或**1300**到**1999**。**

deny/permit**：如果完全匹配，则拒绝或允许该数据包转发。**

source**：源**IP**有三种输入方式：**

A.B.C.D wildcard **可以控制来自一个网段的**IP**地址；**

any **相当于**A.B.C.D 255.255.255.255

host A.B.C.D**相当于**A.B.C.D 0.0.0.0

wildcard**：决定哪些**bits**需要匹配，**'0'**表示需要匹配，**'1'**表示不需要匹配。**

扩展IP**规则：扩展**IP**规则是标准**IP**规则的扩展，可以通过源**IP**，目的**IP**，**IP**协议类型和服务端口来控制数据包的转发。**

命令形式：access-list <groupId> {deny | permit} <protocol> <source> [eq <srcPort>] <destination> [destPort]

参数说明：

groupId **：访问控制列表组号，扩展**IP ACL **支持从**100**到**199**组或**2000**到**2699**。**

deny/permit**：如果完全匹配，则拒绝或允许该数据包转发。**

protocol **:在**IP**层之上的协议类型，如：**icmp**，**tcp**，**udp**等，也可以输入相应的数字**6(tcp)**。如果不需要对这些协议进行控制，可以输入**ip**或**0**。**

source**：源**IP**有三种输入方式：**

1)A.B.C.D wildcard **可以控制来自一个网段的**IP**地址；**

2)any **相当于**A.B.C.D 255.255.255.255

3)host A.B.C.D**相当于**A.B.C.D 0.0.0.0

srcPort **：是对于**protocol**为**tcp**或**udp**的情况，可以控制数据包的源端口，输入方式可以是一些熟悉的端口服务名称，如：**www**也可以是数字，如**80**。**

destination：目的IP有三种输入方式：

1) A.B.C.D wildcard 可以控制来自一个网段的IP地址；

2) any 相当于A.B.C.D 255.255.255.255

3) host A.B.C.D相当于A.B.C.D 0.0.0.0

destPort：是对于protocol为tcp或udp的情况，可以控制数据包的目的端口，输入方式和srcPort相同。

其他命令列表：

show access-list [groupId]

显示当前ACL中配置的规则列表。如果输入了groupId则当前组的规则列表；否则显示所有的规则列表。

no access-list <groupId>

删除指定的规则列表。groupId组的所有规则。

8.4 ACL 过滤配置

交换机缺省所有的端口都没有做ACL过滤。

命令列表：

access-group <groupId>

模式：二层接口配置模式

参数：

groupId：和端口绑定的ACL组号

功能：配置ACL端口过滤。

注意：如果上面的命令配置失败或无效，可能有下面的原因：

ACL组中的规则太多或FFP被QoS 等其它应用占用。

显示ACL端口过滤配置

show acccess-group

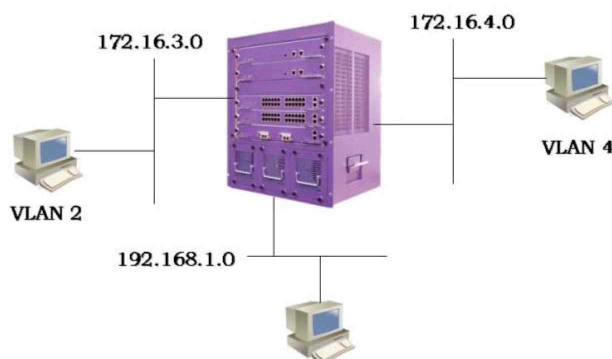
删除当前端口和ACL端口过滤相关的配置

no acl-filter <groupId>

8.5 ACL 配置示例

1、配置

一个交换机连接三个子网，设计ACL， **阻塞源地址为**192.168.1.0 **网络地址。而允许其他网络地址的通信流量通过。**192.168.1.0 **网段连接到**12800 **的**1/1 **端口。**



在交换机上配置如下：

```
Switch#config t
Switch(config)#vlan database
Switch(config-vlan)#vlan 2
Switch(config-vlan)#vlan 3
Switch(config)#interface ge1/1
Switch(config-ge1/1)#swithport mode access
Switch(config-ge1/1)#switchport access vlan 2
Switch(config)#interface vlan2
Switch(config-vlan2)#ip add 192.168.1.1/24
Switch(config)#interface ge1/2
Switch(config-ge1/2)#swithport mode access
Switch(config-ge1/2)#switchport access vlan 3
Switch(config)#interface vlan3
Switch(config-vlan2)#ip add 172.16.3.1/24
Switch(config)#access-list 10deny 192.168.1.0 0.0.0.255
Switch(config)#access-list 10 permit any
Switch(config)#interface ge1/2
Switch(config-ge1/1)#access-group 10
Switch(config-ge1/1)#exit
```

Switch(config)#interface ge1/2
Switch(config-ge1/2)#access-group 10

2、排错

在配置访问控制列表之前确定所有ip **之间都是通的，然后再添加访问控制列表。这条访问控制列表阻塞的是源地址为**192.168.1.0 **网段的**IP **数据流通过交换机。注意子网反码的写法。用**show access-list **命令列出访问控制列表进行查看，一定要注意源地址和目的地址不要写反。然后进行访问控制列表的查看。而且默认访问控制列表最后都有一条隐含的**deny any **的语句，如果想让其他都通过的话，需要添加一条**permit any **的语句，否则都不能够通过。**

第9章 配置IP路由

路由是三层交换机的重要功能之一，能够实现在不同的IP网段间的数据转发。路由的概念很宽泛，以三层接口和ARP协议为基础，包括静态路由和动态路由。静态路由是用户手工配置的路由，而动态路由是通过动态路由协议自动学习到的路由。动态路由协议在后面的章节中介绍，这里只介绍三层接口，ARP 和静态路由的配置。

本章主要包括以下内容：

- 配置VLAN接口
- 配置ARP
- 配置静态路由
- IP 路由配置示例

9.1 配置 VLAN 接口

在交换机中，每个三层接口都是依附在某个VLAN之上的，所以三层接口又称为VLAN接口。VLAN接口的创建和删除是用户在创建和删除VLAN时自动完成的，当用户创建一个VLAN时，系统自动创建与该VLAN对应的VLAN接口，当用户删除一个VLAN时，系统自动删除与该VLAN对应的VLAN接口。交换机支持512个VLAN接口，对于4K个VLAN，只能在512个VLAN上创建VLAN接口。

每个VLAN接口都有一个名称，VLAN接口的名称是字符串"vlan"后接VLAN ID号，如VLAN 1的三层接口的名称为"vlan1"，VLAN 4094的三层接口的名称为"vlan4094"。

与端口一样，VLAN接口也有管理状态和链路状态。目前交换机不提供VLAN接口的管理状态的配置，只要VLAN接口创建好了，VLAN接口的管理状态总是UP。VLAN接口的链路状态是与该接口对应的VLAN所包含的端口相关的，只要VLAN内的一个端口的链路状态是RUNNING，则该VLAN接口的链路状态是RUNNING，如果VLAN内所有端口都不是RUNNING，则该VLAN接口的链路状态也不是RUNNING。

在VLAN接口上可以配置IP地址并指明与此接口相连的网段的网络前缀（可转换为网络掩码）。目前交换机只支持一个VLAN接口上配置一个IP地址。在配置IP地址之前用户需要先创建VLAN并把相关的端口加入到VLAN中。缺省情况下交换机存在VLAN1的接口，并且在此接口上设置了IP地址192.168.0.1/24，用户也可以修改VLAN1接口的IP地址。除VLAN1以外的其它VLAN的接口缺省没有设置IP地址。

配置VLAN接口的IP地址的命令如下表：

命令	描述	CLI模式
ip address <ip-prefix>	在VLAN接口上设置IP地址。参数包括接口的IP地址和相连网段的网络前缀。如果该VLAN接口原来存在IP地址，先删除原来的IP地址，再设置指定的IP地址。参数的格式为A.B.C.D/M。	接口配置模式
no ip address [ip-prefix]	删除VLAN接口的IP地址。如果指定了参数，该参数必须与设置时给定的参数相同，否则此命令无效。参数的格式为A.B.C.D/M。	接口配置模式

查看VLAN**接口的命令如下表：**

命令	**描述**	CLI**模式**
show interface [if-name]	**查看**VLA N**接口的信息，包括接口的**IP**地址，**MAC**地址，管理状态，链路状态等。参数是**VLAN**接口的接口名，如果没有指定参数，则查看所有的端口和**VLAN**接口的信息。**	**普通模式，特权模式**
show running-config	**查看系统的当前配置，可以查看到**VLAN**接口的配置。**	**特权模式**

例子：

在VLAN3**接口上配置子网**193.1.1.0**，子网前缀为**24(**也就是掩码**255.255.255.0)**，接口的**IP**地址为**193.1.1.1**，并且查看**VLAN3**接口的信息。命令如下：**

```
switch(config)#interface vlan3
switch(config-vlan3)#ip address 193.1.1.1/24
switch(config-vlan3)#end
switch#show interface vlan3
```

9.2 配置 ARP

ARP（Address Resolution Protocol）**协议是为**IP **地址到对应的**MAC**地址提供映射的协议。当源端把以太网数据帧发送到位于同一**VLAN**内的目的端时，是根据**48**位的以太网**MAC**地址来确定目的地的，目的端根据数据包的目的**MAC**地址来决定是否需要接收此数据包。**

假定两个相邻网段的主机A**和**B**通过**iSpirit 12800**系列交换机进行通信，主机**A**在发送数据给主机**B**之前，首先向与主机**A**直接相连的交换机的接口发出**ARP **请求报文，得到**ARP **应答后发送数据包到该接口。交换机收到此数据包后首先向主机**B**广播一个**ARP**请求报文，从主机**B**处得到**ARP **响应报文后**,**再把数据包发送给主机**B。

交换机上有一个ARP高速缓存，称为ARP表，存放直接相连的网络中的IP地址到MAC地址的映射记录。ARP表中每一项都有一个生存时间，缺省是20分钟，当交换机在生存期间内没有收到该IP地址的ARP请求或应答报文，则该IP地址对应的ARP表项将被删除。

本节包括以下内容：

- 配置静态ARP
- 配置ARP绑定
- 查看ARP的信息

9.2.1 配置静态 ARP

在ARP表中存在两种不同的ARP表项，一种是静态ARP，一种是动态ARP。静态ARP是用户通过命令配置的ARP表项，系统不会自动刷新和删除，需要用户手工完成。动态ARP是系统根据收到的ARP请求或应答包自动学习到的ARP，系统自动创建和删除，实时更新和维护，不需要用户干预，但用户可以手工删除动态ARP表项。

交换机缺省没有配置静态ARP表项。需要注意的是当某个VLAN接口被删除或接口的子网网段IP改变时，原来的子网网段内的静态和动态ARP表项都被删除。

配置静态ARP的命令如下表：

命令	描述	CLI模式
arp <ip-address> <mac-address>	配置静态ARP表项。第一个参数是IP地址，IP地址必须在某个子网网段内。第二个参数是MAC地址，MAC地址必须是单播MAC地址，MAC地址的格式为HHHH.HHHH.HHHH，如0010.5cb1.7825。	全局配置模式
no arp {<ip-address> | <ip-prefix> | all | dynamic | static}	删除ARP表项。包括删除一个IP的ARP表项；删除一个网段的ARP表项；删除所有的ARP表项；删除所有的动态ARP表项，删除所有的静态ARP表项。	全局配置模式

9.2.2 配置 ARP 绑定

ARP绑定是为了增强网络的安全性考虑的。对于某个子网网段，只允许规定的IP地址和其对应的MAC地址的主机可以访问网络，防止非法用户使用网络。当某个接口的子网网段做了ARP绑定后，该接口不学习动态ARP 表项，该接口的所有ARP表项都是静态的，所有的ARP 数据包都是根据静态ARP表项来处理的。

配置ARP绑定有下面几种方法：

第一种方法：先配置接口子网网段内的静态ARP表项，再给接口子网网段上锁。

第二种方法：先给接口子网网段上锁，再配置接口子网网段内的静态ARP 表项。

第三种方法：先把接口子网网段内的所有动态ARP表项修改为静态ARP表项，再给接口子网网段上锁。这种方法对于用户来说使用非常方便。

接口子网网段缺省没有配置ARP绑定。

配置ARP绑定的相关命令如下表：

命令	描述	CL模式
arp static {<ip-prefix> | all}	把某个网段内的或所有的动态ARP 表项修改为静态ARP表项。	全局配置模式
arp lock <ip-prefix>	给某个网段上锁，上锁后该网段不再学习动态ARP表项，并且删除以前学习到的该网段内的所有动态ARP表项。	全局配置模式
arp unlock {<ip-prefix> | all}	给某个或所有的已经上锁的网段解锁，解锁后网段可以学习动态ARP 表项。	全局配置模式

例子：

对网段193.1.1.0/24进行ARP 绑定，只允许IP 地址为193.1.1.100，MAC地址为0010.5cb1.7825的主机和 IP地址为193.1.1.101，MAC地址为0010.5cb1.7826的主机访问网络。

如果在配置ARP 绑定时在ARP表中不存在IP地址为193.1.1.100和193.1.1.101的ARP表项，配置如下：

Switch(config)#arp lock 193.1.1.0/24

Switch(config)#arp 193.1.1.100 0010.5cb1.7825

Switch(config)#arp 193.1.1.101 0010.5cb1.7826

如果在配置ARP 绑定时在ARP表中存在IP地址为193.1.1.100和193.1.1.101的ARP 表项，配置如下：

Switch(config)#arp static 193.1.1.0/24

Switch(config)#arp lock 193.1.1.0/24

9.2.3 查看 ARP 的信息

查看ARP的信息的命令如下表：

命令	描述	CL模式
show arp [<ip-prefix> | dynamic | static]	查看ARP 表中的ARP 表项信息，包括所有的ARP表项，某个网段的ARP表项，动态ARP表项和静态ARP表项。	普通模式，特权模式
show arp lock	查看ARP 绑定的信息。	普通模式，特权模式
show running-config	查看系统的当前配置，可以查看到ARP 的配置。	特权模式

9.3 配置静态路由

静态路由是由用户定义的、一条可使数据包从源地址通过指定路径到达目的地址的路由。当动态路由协议未能创建一条到特定目的的路由时，静态路由就显得特别重要。还可以通过配置某一静态路由为缺省路由，把无法确定路由的数据包发送到默认的网关。

静态路由是由管理员手工配置而成。适用于组网结构较简单的网络，管理员只需配置静态路由就能使交换机正常工作。静态路由由于不会有路由更新而不会占用宝贵的网络带宽。

缺省路由也是一种静态路由。简单地说，缺省路由就是在没有找到任何匹配的路由项情况下，才使用的路由。即只有当无任何合适的路由时，缺省路由才被使用。在路由表中，缺省路由以到网络0.0.0.0/0 （掩码为0.0.0.0） 的路由形式出现。若报文的目的地不在路由表

中且路由表中也无缺省路由存在，该报文被丢弃的同时将返回源端一个ICMP 报文指出该目的地址或网络不可达信息。缺省路由在网络中是非常有用的。在一个包含上百个交换机的典型网络中，运行动态路由选择协议可能会耗费较大量的带宽资源，使用缺省路由就可节约因路由选择所占用的时间与包转发所占用的带宽资源，这样就能在一定程度上满足大量用户同时进行通信的需求。

交换机可以配置多条到同一目的地的静态路由，但只有其中的一条路由生效，用于实际的数据转发。交换机缺省没有配置静态路由。

配置静态路由的命令如下表：

命令	描述	CLI模式
ip route <ip-prefix> <nexthop-address>	设置静态路由。第一个参数指定网段IP和网络前缀，第二个参数指定下一跳。	全局配置模式
ip route <ip-address> <mask-address> <nexthop-address>	功能与上一个命令相同。第一个参数指定网段的IP地址，第二个参数指定网段的掩码，第三个参数指定下一跳。	全局配置模式
no ip route <ip-prefix> [nexthop-address]	删除静态路由。第一个参数指定网段IP和网络前缀，第二个参数指定下一跳。如果没有第二个参数，则删除与指定网段匹配的所有路由。如果有第二个参数，则删除与指定网段和下一跳都匹配的路由。	全局配置模式
no ip route <ip-address> <mask-address> [nexthop-address]	功能与上一个命令相同。第一个参数指定网段的IP地址，第二个参数指定网段的掩码，第三个参数指定下一跳。如果没有第三个参数，则删除与指定网段匹配的所有路由。如果有第三个参数，则删除与指定网段和下一跳都匹配的路由。	全局配置模式

查看路由的命令如下表：

命令	描述	CL模式
show ip route [<ip-address> | <ip-prefix> | connected | static | rip | ospf]	**查看正生效的路由的信息，包括所有的路由，某个路由，某个网段的路由，直接相连的路由、静态路由、**RIP **路由和**OSPF**路由。**	**普通模式，特权模式**
show ip route database [connected | static | rip | ospf]	**查看正生效和没生效的路由的信息，包括所有的路由，直接相连的路由、静态路由、**RIP **路由和**OSPF**路由。**	**普通模式，特权模式**
show running-config	**查看系统的当前配置，可以查看到静态路由的配置。**	**特权模式**

例子：

设置目的IP**地址为**200.1.1.0**，子网掩码为**255.255.255.0**，下一跳为**10.1.1.2**的静态路由。配置命令为：**

Switch(config)#ip route 200.1.1.0 255.255.255.0 10.1.1.2

或 Switch(config)#ip route 200.1.1.0/24 10.1.1.2

删除目的IP**地址为**200.1.1.0**，子网掩码为**255.255.255.0**，下一跳为**10.1.1.2 **的静态路由。配置命令为：**

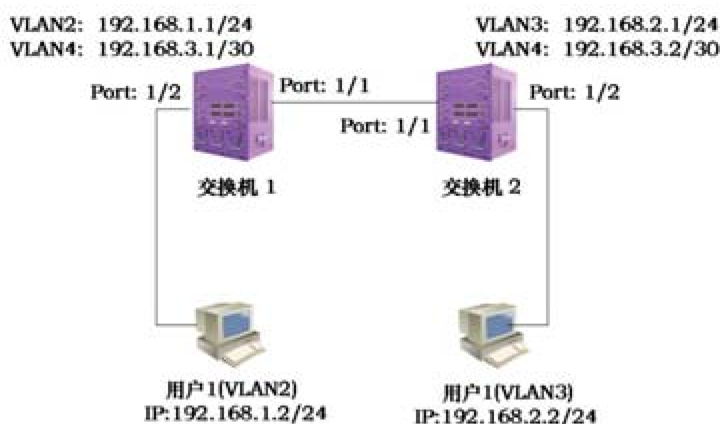
Switch(config)#no ip route 200.1.1.0/24

或 Switch(config)#no ip route 2001.1.0/24 10.1.1.2

或 Switch(config)#no ip route 200.1.1.0 255.255.255.0

或 Switch(config)#no ip route 200.1.1.0 255.255.255.0 10.1.1.2

9.4 IP 路由配置示例



9.4.1 三层接口

在交换机1**上配置**VLAN2**对应的三层接口，同时分配一个**IP**地址**192.168.1.1/24。

配置如下：

```
Switch#config t
Switch(config)#vlan database
Switch(config-vlan)#vlan 2
Switch(config-vlan)#exit
Switch(config)#interface ge1/2
Switch(config-ge1/2)#switch access vlan 2
Switch(config)#interface vlan2
Switch(config-vlan2)#ip add 192.168.1.1/24
```

验证：用户1**能够**ping**通交换机**1**的**VLAN2**对应的三层接口**IP**地址。**

9.4.2 静态路由

要求通过配置静态路由，使两个用户PC**之间能够互通。**

交换机1上配置如下：

```
Switch#config t
Switch(config)#ip route 192.168.2.0 255.255.255.0 192.168.3.2
```

交换机2上配置如下：

```
Switch#config t
Switch(config)#ip route 192.168.1.0/24 192.168.3.1
```

验证：用户1**能**ping**通用户**2**，用户**2**能**ping**通用户**1。

9.4.3 ARP

把交换机1**的**VLAN2**进行**ARP**锁定，配置用户**1**的静态**ARP**，只允许用户**1**从**VLAN2**接入。假定用户**1**的**MAC**地址是**00:00:00:00:00:01。

交换机1配置如下：

```
Switch#config t
Switch(config)#arp lock 192.168.1.0/24
Switch(config)#arp 192.168.1.2 0000.0000.0001
```

验证：用户1**能够**ping**通交换机**1**的**VLAN2**对应的三层接口**IP**地址。如果把用户**1**的**IP**地址改成**192.168.1.3**，则**ping**不通交换机**1**的**VLAN2**的接口**IP**地址。**

第10章 配置RIP

本章主要包括以下内容：

- RIP **介绍**
- RIP **配置**
- RIP **配置示例**

10.1 RIP 介绍

RIP（路由信息协议 Routing Information Protocol）是较早开发的动态路由协议，使用距离向量算法，多应用在小型网络中。RIP协议报文封装在UDP报文中，使用UDP端口520。RIP的主要思想是使用跳数（hop）来衡量到达宿主机的距离，每经过一台路由器跳数增加1，以此来计算路由的权值（metric），进行选路。RIP约定最大跳数为15，跳数16标记为网络不可达。RIP使用广播整个路由表的方式让网络的中的路由器同步路由信息，每隔30秒定时更新一次报文，若某条路由条目在180秒内未收到从邻居发来的更新报文，则将其标记为不可达，如果再过120秒还未接收到有效更新，则将该路由删除。

RIP因其简单的思想容易实现，但也带来了相应的路由环路问题，为防止路由环，RIP引入了水平分割机制来规避路由器之间的欺骗。水平分割也即路由更新不会从接收到的接口发布出去。带毒性反转的水平分割是将路由更新从接收到的接口发布出去，但权值标记为不可达，这样能让邻居路由器快速识别环路而无需等待权值增加到不可达。

路由表中的路由条目应包含目的地址（主机或网络）下一跳地址、转发接口、路由权值、计时器（当接收到路由更新时计时器被重置）路由标记。

RIP启动时，立即以广播（RIP-1）或组播（RIP-2）形式发送一全表请求报文，相邻路由器接收到请求报文就会将自己完整的路由表以响应报文回送。路由器接收到响应报文会逐条处理路由，并修改自己的路由表，当有新路由时会立即产生一触发更新报文。经过一连串的更新过程，最终RIP收敛，网络中各个路由器均保持了最新的一致的路由信息。网络稳定后，RIP依然每隔30秒向邻居广播本地路由表，各个路由器根据接收到的路由更新报文维护自身的路由信息，进行最优选路。RIP使用超时机制处理久未更新的路由条目，以保证路由的实时正确。

RIP多应用于校园网及结构简单的较连续的区域性网络，复杂的大型网络RIP难以胜任。

10.2 RIP 配置

启动RIP协议后可进行RIP各功能各属性的配置。RIP的配置多在RIP配置模式和接口配置模式下。

RIP的配置包括：

- 启动 RIP 并进入 RIP 配置模式

- 使能 RIP 接口
- 配置单播报文传送
- 配置接口的工作状态
- 配置缺省路由权值
- 配置管理距离
- 配置计时器
- 配置版本
- 引入外部路由
- 配置路由过滤
- 配置附加路由权值
- 配置接口的 RIP 版本
- 配置接口的收发状态
- 配置水平分割
- 报文认证
- 配置接口权值

10.2.1 启动 RIP 并进入 RIP 配置模式

模式：全局配置模式

命令：router rip

启动rip并进入rip配置模式

命令：no router rip

关闭rip协议

缺省：不运行rip协议

10.2.2 使能 RIP 接口

RIP工作时可指定某些接口，将其所在的网络配置成RIP 网络，即可在其上收发RIP 协议报文。

模式：RIP 配置模式

命令：network <network-address>

使能rip**接口**

命令：no network <network-address>

关闭rip**接口**

参数：有A.B.C.D/M**和**A.B.C.D A.B.C.D**两种形式，前一种指定网络**ip**及掩码长度，后一种指定网络**ip**及掩码。**

缺省：RIP **协议启动后在所有接口禁用**

RIP**协议启动后必须指定其工作的网段，**RIP **只能在指定网段的接口上运行，对于那些不在指定网段的接口，**RIP**既不接收发送路由也不会将接口路由转发。在**RIP**的视图中，那些不在指定网段的接口不存在。参数**network-address**为使能或者不使能的网络地址，可配置为接口**ip**地址。**network**命令使能该地址的网段的接口。例如：有一接口的**ip**地址为**192.160.1.1**，使用命令**network 192.160.1.1/24**，使用**show running-config**命令看到的是**network 192.160.1.0/24。

10.2.3 配置单播报文传送

RIP**协议版本**1**使用广播交换报文，版本**2**使用组播（**224.0.0.9**）交换报文，当在不支持广播的链路上运行**RIP**协议时，需要指定特定的单播地址来交换报文。**

模式：RIP **配置模式**

命令：neighbor <ip-address>

配置对端单播ip**地址**

命令：no neighbor <ip-address>

取消对端单播ip**地址的设置**

参数：ip-address**为指定的单播**ip**地址**

缺省：RIP **协议不向任何单播地址发送报文**

10.2.4 配置接口的工作状态

RIP**协议运行在某些网络中，可能仅需要**RIP **接口路由，并不希望在该接口上广播**RIP **路由。使用**network**命令可指定该接口上收发**RIP **协议报文，且可以获知该接口路由。使用**

passive-interface命令仅获知该接口路由而阻塞该接口的广播。

模式：RIP 配置模式

命令：passive-interface <if-name>

配置接口为被动状态

命令：no passive-interface <if-name>

取消接口被动状态

参数：if-name为约定的三层接口名（例如：vlan1 vlan2 …）

缺省：使能的RIP接口均不为passive 状态

10.2.5 配置缺省路由权值

在引入外部路由时，需要指定一个路由权值；当未指定其路由权值时，使用这个缺省路由权值。

模式：RIP 配置模式

命令：default-metric <metric>

设置引入外部路由时缺省路由权值

命令：no default-metric [metric]

恢复引入外部路由时缺省路由权值为默认值1

参数：metric取值在1~16之间，大于1，小于16。

缺省：metric值为1，使用no default-metric命令恢复到缺省值。

10.2.6 配置管理距离

每一种协议都有约定的优先级，管理距离即使用路由策略时选择路由的优先级。当存在到达同一目的地的两条相同路由（来自不同的路由协议），则管理距离越小，优先选择该协议的路由。

模式：RIP 配置模式

命令：distance <distance>

设置管理距离值

命令：no distance [distance]

恢复管理距离为缺省值

参数：distance**取值在**1~255**之间**

缺省：distance**值为**120，**使用**no distance**命令恢复到缺省值。**

10.2.7 配置计时器

RIP**协议有三个计时器，其一是完整路由表每**30**秒向所有**RIP**接口广播一次，其二是**RIP**路由表中每条路由若**180**秒未接收到更新则标记**metric**为**16，**其三是**RIP**路由表中每条路由若标记**metric**为**16**后又**120**秒未被有效更新则从路由表中删除。**

模式：RIP **配置模式**

命令：timers basic <update> <timeout> <garbage>

设置三个计时器值

命令：no timers basic

恢复计时器为缺省值

参数：第一个参数update**为整个**RIP**路由表定时更新计时器，第二个参数**timeout**为每条路由超时未更新计时器，第三个参数**garbage**为每条路由标记为无效后超时需删除计时器；三个计时器的取值范围均为**5~(2^{31}-1)。

缺省：update**为**30**秒更新一次；**timeout**为**180**秒标记为无效；**garbage**为**120**秒删除。**

10.2.8 配置版本

RIP**协议目前有版本**1（RFC1058）**和版本**2（RFC2453），**配置的版本值会体现在协议报文的版本域中。**

模式：RIP **配置模式**

命令：version <version>

设置RIP **协议为版本**1**或者版本**2

命令：no version [version]

恢复RIP**协议版本为缺省值**

参数：version**可取值**1**或者**2

缺省：版本2

10.2.9 引入外部路由

RIP**允许用户将其他协议的路由信息引入到**RIP**的路由表中，**RIP**可引入的路由协议（类型）包括：**connected、static、OSPF、IS-IS、BGP。

模式：RIP **配置模式**

命令：redistribute {kernel | connected | static | ospf | isis | bgp} [metric <metric> | route-map <route-map-name>]

引入其他协议路由

命令：no redistribute {kernel | connected | static | ospf | isis | bgp} [metric <metric> | route-map <route-map-name>]**取消引入的路由**

参数：第一个参数为引入其他协议的名称，可引入的有直连、静态、ospf、is-is、bgp**；第二个参数为引入时设置的权值，取值**1~16**之间；第三个参数为引用的**route-map**名称，**route-map**在全局配置模式下配置，可参看命令手册。**

缺省：RIP **协议不引入任何外部协议**

10.2.10 配置路由过滤

RIP**提供路由过滤功能，通过指定的访问控制列表和地址前缀列表，对接收到的路由和发布的路由，配置策略规则进行过滤。**

模式：RIP **配置模式**

命令：distribute-list <acl-name> {in | out} [if-name]

使用access-list**过滤接口的输入输出**

命令：no distribute-list <acl-name> {in | out} [if-name]

取消使用access-list**过滤**

参数：acl-name**表示引用的**access-list **的名；**if-name**表示应用到的**RIP**接口；**in**和**out

表示应用在接收到路由的方向还是发布路由的方向上。

命令：distribute-list prefix <pre-name> {in | out} [if-name]

使用prefix-list**过滤**

命令：no distribute-list prefix <pre-name> {in | out} [if-name]

取消使用prefix-list**过滤**

参数：pre-name**表示引用的**prefix-list**的名；**if-name**表示应用到的**RIP**接口；**in**和**out**表示应用在接收到路由的方向还是发布路由的方向上。**

缺省：RIP **协议不对任何接收和发送的路由进行过滤**

access-list**和**prefix-list**在全局配置模式下配置，可参考命令手册。**

10.2.11 配置附加路由权值

附加路由权值是对RIP**协议的路由权值在输入输出时添加的一个偏移量值，并不直接改变路由表中路由的权值，而是在接口接收发送路由时增加一个偏移量。**

模式：RIP **配置模式**

命令：offset-list <acl-name> {in | out} <offset> [if-name]

使用access-list**对接口输入输出路由的权值增加一偏移量**

命令：no offset-list <acl-name> {in | out} <offset> [if-name]

取消输入输出路由的权值的偏移量

参数：acl-name**表示引用的**access-list**名；**in**和** out**表示应用在输入还是输出方向上；**offset**表示偏移量的值，取值**0~16**之间；**if-name**表示应用到的**RIP**接口。**

缺省：在接收报文时每条路由的附加权值为1**，在发送报文时每条路由的附加权值为**0。

10.2.12 配置接口的 RIP 版本

RIP**分**RIP-1**和**RIP-2**两个版本，可以对使能的**RIP **协议的接口指定其处理的**RIP **报文版本。接收方向，可区分为仅接收**RIP-1**的报文，仅接收**RIP-2**的报文，既接收**RIP-1**又接收**RIP-2**的报文。在发送方向上，可区分为发送**RIP-1**的报文，发送**RIP-2**的报文（以广播方式），发送**RIP-2**的报文（以组播方式），即发送**RIP-1**又发送**RIP-2**的报文。**RIP-2**有广播和组播两种发送报文方式，使用组播既可以避免同一网络中没有运行**RIP**的主机不接收**RIP **的广播报文，又可以避免运行**RIP-1**的主机错误的处理**RIP-2**的带有子网掩码的路由。**

模式：接口配置模式

命令：ip rip receive version {1 | 2}

设置接口仅接收版本1的报文或者仅接收版本2的报文

参数：版本1**或者版本**2

命令：ip rip receive version {1 2 | 2 1}

设置接口既可以接收版本1的报文又可以接收版本2的报文

参数：可以写作1 2**或者**2 1

命令：no ip rip receive version [1 | 2 | 1 2 | 2 1]

恢复接口接收报文设置为缺省值

缺省：版本2**组播方式**

命令：ip rip send version {1 | 2 | 1-compatible}

设置接口仅发送版本1的报文或者仅发送版本2的报文

参数：版本1**或者版本**2**；**1-compatible**表示版本**2**的接口发送出兼容版本**1**的报文，也即广播报文而非组播。**

命令：ip rip send version {1 2 | 2 1}

设置接口既可以发送版本1的报文又可以发送版本2的报文

参数：可以写作1 2**或者**2 1

命令：no ip rip send version [1 | 2 | 1-compatible | 1 2 | 2 1]

恢复接口发送报文设置为缺省值

缺省：版本2**组播方式**

10.2.13 配置接口的收发状态

在RIP**模式下使用**network**命令使能了**RIP**接口后，还可以在接口模式下指定其收发协议报文的状态，是否接收协议报文或者是否发送协议报文。**

模式：接口配置模式

命令：ip rip receive-packet

配置接口接收协议报文

命令：no ip rip receive-packet

配置接口不接收协议报文

命令：ip rip send-packet

配置接口发送协议报文

命令：no ip rip send-packet

配置接口不发送协议报文

缺省：使能接收发送协议报文

注意区别，network**命令启动某一网络运行**RIP**协议，在该网络内的接口收发协议报文，该接口路由包含在路由表中。**passive-interface**命令是在**network**命令生效后，使该接口不收发协议报文，但该接口路由仍包含在路由表中。**ip rip receive-packet**和**ip rip send-packet **命令也是在**network**命令生效后，具体指定接口是否接收或者是否发送协议报文。**

10.2.14 配置水平分割

水平分割是指从本接口接收到的路由不从本接口发出。带毒性反转的水平分割是指从本接口接收到的路由依然从本接口发出，但其metric**值标记为**16**。水平分割可以在一定程度上避免产生环路，带毒性反转的水平分割比普通水平分割效率更高，直接标记不可达。但在**NBMA**网络上需要禁止水平分割来获取正确的路由。**

模式：接口配置模式

命令：ip rip split-horizon [poisoned]

启动接口水平分割功能或带毒性反转的

命令：no ip rip split-horizon

禁止接口的水平分割功能

参数：无poisoned**参数表示启动普通水平分割功能，带**poisoned**参数表示启动带毒性反转的水平分割功能。**

缺省：带毒性反转的水平分割

10.2.15 报文认证

RIP-1**不支持报文认证，**RIP-2**支持报文认证，有两种认证方式，明文认证和**MD5**认证。明文认证中未加密的认证数据随报文一同传送，不能提供安全保障，不能应用于安全性要求**

较高的网络。密码的设置分普通密钥及密钥链两种，普通密钥保存独立的字符串，密钥链管理密钥的id**、内容、接收的生存期、发送的生存期。密钥链管理详见命令参考手册。**

模式：接口配置模式

命令：ip rip authentication mode {text | md5}

设置认证模式明文或者md5

命令：no ip rip authentication mode [text | md5]

取消认证

参数：text**为明文认证，**md5**认证。**

缺省：无认证

命令：ip rip authentication string <password>

设置认证的密码串

命令：no ip rip authentication string [password]

取消认证的密码串

参数：16**个字节的认证密码**

命令：ip rip authentication key-chain <key-chain-name>

设置认证的key-chain

命令：no ip rip authentication key-chain [key-chain-name]**取消认证的**key-chain

参数：引用的key-chain**的名；**key-chain**在全局配置模式下配置，参看命令手册。**

10.2.16 配置接口权值

模式：接口配置模式

命令：ip rip metric <metric>

配置接口权值

命令：no ip rip metric

恢复接口权值为默认值

参数：metric**取值**1-16**之间，表示该接口学习到的路由条目需增加的权值。**

缺省：为1

10.2.17 显示信息

模式：普通模式或特权模式

命令：show ip protocols

显示所有运行中的协议的信息

命令：show ip protocols rip

显示RIP **协议信息**

命令：show ip rip

显示RIP **路由**

命令：show ip rip database

显示RIP **数据库**

命令：show ip rip database count

显示RIP **数据库条目数**

命令：show ip rip interface [if-name]

显示RIP **接口信息**

参数：if-name**为约定的三层接口名**

模式：特权模式

命令：show running-config

显示交换机当前配置，包括RIP **配置。**

命令：show running-config rip

显示RIP **协议的当前配置。**

10.3 RIP 配置示例

(1) 配置

三台交换机两两相连，分别有6 **个网段，都启用**rip **协议，实现三台**PC **机之间能够两两互通。**

在交换机1**上：**

Switch# config t

Switch(config)#router rip

Switch(config-rip)#network 192.168.1.0/24

Switch(config-rip)#network 10.1.1.0/24

Switch(config-rip)#network 10.1.2.0/24

在交换机2上：

Switch# config t

Switch(config)#router rip

Switch(config-rip)#network 192.168.2.0/24

Switch(config-rip)#network 10.1.2.0/24

Switch(config-rip)#network 10.1.3.0/24

在交换机3上：

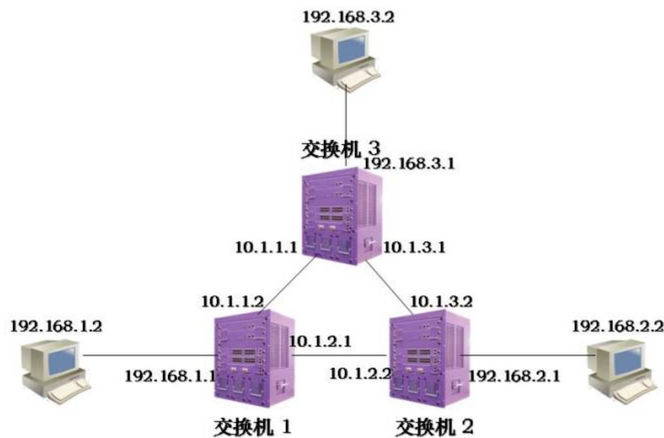
Switch# config t

Switch(config)#router rip

Switch(config-rip)#network 192.168.3.0/24

Switch(config-rip)#network 10.1.1.0/24

Switch(config-rip)#network 10.1.3.0/24



(2) 验证

使用下面的命令查看RIP的信息：

show ip protocols rip

show ip rip database

show ip rip interface

第11章 配置OSPF

本章主要包括以下内容：

- OSPF **介绍**
- OSPF **配置**
- OSPF **配置示例**

11.1 OSPF 介绍

OSPF（开放最短路径优先协议 Open Shortest Path First）是基于链路状态算法的协议，可支持较大规模网络，收敛速度较快。

运行 OSPF 协议的路由器各自维护链路状态数据库（LSDB），该数据库描述整个自治系统的拓扑结构，仿佛一张地图。当所有路由器的数据库同步后，每台路由器以自身为视角，计算出到达自治系统中其他目的结点的最短路由，维护在自己的路由表中。当网络中拓扑发生变化时，路由器只需要将变化的链路状态封装在链路状态更新（LSU）报文中广播出去，所有的路由器会再次同步本地的数据库，重新计算路由。每台路由器将自己看到的链路状态广播（LSA）都发布出去，集中起来，就形成了整个网络的拓扑描述 LSDB，将其转换为带权的有向图，就可以使用 SPF 算法计算出路由表。

在广播网络中每台路由器都需要将各自的状态信息广播到其他路由器,就会建立多个两两邻接关系，这会带来大量没有必要的报文传送。为此，OSPF 约定了指定路由器（DR）和备份指定路由器（BDR），路由器将链路信息发给 DR，由 DR 收集整理再发给所有的路由器。有效的减少了广播网络上路由器之间的邻接数量。

OSPF 支持五种协议报文：

HELLO 报文，周期性的广播给邻居，用于发现和维护邻居，进行 DR 选举，包含一些接口属性值，HELLO 报文中的一些参数必须一致方能建立邻居。

DD 报文（Database Description）在同步过程中使用 DD 报文来描述自己的 LSDB，包括每一条 LSA 的 head，通过 LSA head 可以唯一确定一条 LSA，对端路由器可以判断自己是否有这条 LSA；若无，再请求完整的 LSA。

LSR 报文（Link State Request）两台路由器交换 DD 报文之后，就会知道对端路由器有哪些 LSA 是本地缺少的，这时需要发送 LSR 报文请求完整的 LSA。请求时只需要 LSA head 即可。

LSU 报文（Link State Update）是多条 LSA 的集合。

LSAck 报文（Link State Acknowledgement）是对接收到的 LSU 报文确认，保证可靠的传递链路信息。使用 LSA head 确认。

Router-id 概念：自治系统内路由器的唯一标识。

区域（area）:OSPF 若运行在一个较大的网络中，由于路由器数量的大增，会导致 LSDB 非常庞大，并使同步的时间及计算路由的时间增加，占用大量存储空间及 CPU 资源。并且越大的网络，拓扑变化越频繁，使得网络经常处于变动中，路由器需要花费大量的时间传递报文计算路由，无谓的占用了网络带宽。因此 OSPF 引入区域概念，将路由器划分在不同

的区域内，LSDB 只在区域内同步并在区域内计算路由，区域之间的路由交互由边界路由器（ABR）来完成。这样在区域内路由器数量会有所限制，LSDB 也会局限于较小的容量，计算路由的时间会大大缩减，当拓扑变化时收敛也很快。区域概念有效的将一个大范围的网络进行分组，在各个区域内部承担小范围的路由功能。区域之间的路由在骨干区域上交互（区域 ID 为 0 的区域）。因此所有非骨干区域必须与骨干区域相连，也即 ABR 至少有一个接口连接骨干区域。若网络规划，有非骨干区域无法与骨干区域连通，则必须配置虚链路建立逻辑上的通路，也即骨干区域上的某一 ABR 与非骨干区域的某一 ABR 通过一传输区域建立点到点的链路。那么骨干区域上的域间路由信息也会通过虚链路发布到该非骨干区域。

11.2 OSPF 配置

OSPF 协议启动后进入 OSPF 配置模式可进行相应属性及功能设置。OSPF 配置命令多在 OSPF 配置模式下及接口配置模式下。

OSPF 的配置包括：

- 启动 OSPF 并进入 OSPF 模式
- 使能接口
- 指定主机
- 配置路由器 ID
- 配置邻接点
- 禁止接口发送报文
- 配置 SPF 计时器
- 配置管理距离
- 引入外部路由
- 配置接口的网络类型
- 配置 hello 报文发送时间间隔
- 配置邻居路由器失效时间
- 配置重传间隔
- 配置接口延时
- 配置接口在 DR 选举中的优先级
- 配置接口上发送报文的代价

- 配置接口发送 DD 报文是否填MTU 值
- 配置接口报文认证
- 配置区域虚链路
- 配置区域路由聚合
- 配置区域报文认证
- 配置 stub 区域
- 配置 nssa 区域
- 配置外部路由聚合
- 配置外部路由的缺省权值

11.2.1 启动 OSPF 并进入 OSPF 模式

OSPF 协议可运行多个副本，使用进程号（process-id）来标识；启动 OSPF 协议时需说明启动的是哪个进程号的进程；若无参数则进程号为 0。

模式：全局配置模式

命令：router ospf [process-id]

启动进程号 process-id 的 OSPF 进程并进入其模式

命令：no router ospf [process-id]

关闭进程号 process-id 的 OSPF 进程

参数：process-id 取值 1~(2^{16}-1)之间，表示启动的 OSPF 进程号 若不带参数 process-id 则启动进程号为 0 的 OSPF。

缺省：不运行 OSPF 协议

11.2.2 使能接口

OSPF 协议的可贵之处在于引入了分层思想，将一个完整的自治系统分成不同的区域，以期建立一个概念上的层次化网络模型。区域是逻辑上的，将自治系统中的路由器人为的分组。当路由器的不同接口属于不同的区域时，也即跨区域时，称其为边界路由器 ABR。对

于每个启动 OSPF 协议的网段只能隶属于某一特定区域，也即路由器上每个运行 OSPF 协议的接口必须属于指定区域。区域使用区域号（area-id）来标识，区域号为 0 的区域为骨干区域。不同区域之间的路由信息通过边界路由器来传递。区别于 RIP 协议，在接口上运行 OSPF 协议时必须指定其所属区域。

模式：OSPF 配置模式

命令：network <network-address> area <area-id>

指定区域指定接口运行 OSPF 协议

命令：no network <network_address> area <area-id>

关闭特定区域特定接口上的 OSPF

参数：network-address 有 A.B.C.D/M 和 A.B.C.D A.B.C.D 两种形式，前一种指定网络 ip 及掩码长度，后一种指定网络 ip 及掩码。area-id 也有两种形式 A.B.C.D 和整型数，前一种使用点分十进制格式，后一种取值 0~(2^{32}-1)之间。

缺省：OSPF 协议启动后不使能接口

11.2.3 指定主机

模式：OSPF 配置模式

命令：host <ip-address> area <area-id> [cost <cost>]

配置主机路由

命令：no host <ip-address> area <area-id> [cost <cost>] 取消主机路由

参数：ip-address 使用 A.B.C.D 格式，表示某一区域内一指定主机，在路由器的链路表示上是 stub 类型。area-id 同 network 命令中说明。cost 表示指定该链路的代价，为可选参数。

缺省：cost 若不配置则缺省为 0

11.2.4 配置路由器 ID

路由器的 ID **是一个** 32bit **的无符号整数，是一台路由器在自治系统中唯一的标识。路由器** ID **可以手工配置，配置时需保证自治系统内任意两台路由器的** ID **都不相同。若不配置，则路由器使用** loopback **接口的** IP **地址；若** loopback **无** IP **则从当前接口的** IP **地址中选择最高地址作为** ID**。为了保证** OSPF **运行稳定，在网络规划时就应进行路由器** ID **的划分并手工配置。**

模式：OSPF **配置模式**

命令：ospf router-id <router-id>

配置路由器 ID

命令：no ospf router-id

取消路由器 ID

命令：router-id <router-id>

命令：no router-id [router-id]

参数：router-id **使用** A.B.C.D **格式**

缺省：OSPF **协议启动后会根据规则自动生成路由器** ID**。规则如下：首先选用该命令配置的** router-id**；若无，选择** loopback **的** IP **地址；若无，选择当前接口的最高** IP **地址；若无，则为** 0.0.0.0**。**

两组命令功能一样。

11.2.5 配置邻接点

OSPF **协议交互协议报文通过组播方式，使用组播地址** 224.0.0.5 **或者** 224.0.0.6**。当** OSPF **协议运行在不支持广播的链路上，例如** NBMA**，则必须进行一些配置，使用单播方式交互协议报文。这时可手工指定对端的** IP **地址及相应属性值。**

模式：OSPF **配置模式**

命令：neighbor <ip-address> [priority <prio> | poll-interval <deadtime> | cost <cost>]

指定对端邻接点及设置属性

命令：no neighbor <ip-address> [priority <prio> | poll-interval <deadtime> | cost <cost>]**取消对端邻接点及属性设置**

参数：ip-address **对端的** IP **地址，为** A.B.C.D **格式**；prio **为对端优先级，取值** 0~255 **之间**；deadtime **为认为对端** down **的计时，若计时终止则不再向对端发送** hello **报文，取值** 1~(2^{16}-1)**之间**；cost **为到对端链路的代价，取值** 1~(2^{16}-1)**之间。**

缺省：priority **为** 1（**为** 0 **则不参与** DR **选举**）；poll-interval **为** 120 **秒**；cost **为** 10。

11.2.6 禁止接口发送报文

当在一个简单的网络中，OSPF **协议的接口仅表示两台设备之间的一个网段，仅仅是为了传输数据，那么将该接口设置为** passive **状态，阻塞** hello **报文在其链路上广播，这不影响获知该接口路由。**

模式：OSPF **配置模式**

命令：passive-interface <if-name>

配置接口为被动状态

命令：no passive-interface <if-name>

取消接口被动状态

参数：if-name **为三层接口名**（**例如**：vlan1 vlan2…）

缺省：OSPF **协议启动后使能的接口均不为** passive **状态**

将运行 OSPF **协议的接口指定为** passive **状态，该接口的直连路由仍可以发布，但接口上的** OSPF **报文将被阻塞，接口也无法建立邻居关系。在某些组网情况下，可有效节约网络资源。**

11.2.7 配置 SPF 计算时间

当 OSPF **的链路状态数据库** LSDB **发生改变时，需要重新计算最短路径。如果每次改变都立即计算最短路径，将占用大量的资源，并影响路由器的效率。通过配置** delay **和** hold **两个值来调节** SPF **计算的时间间隔，可以抑止网络频繁变化引起的过于频繁的** SPF **计算，从而避免集中在一个时间内占用大量的系统资源，影响路由器运行效率。**

SPF **计算有一计时器，每次按照抑止时间启动下一次的计算。当计时器终止需启动** SPF

计算时，重新计算上一次 SPF **计算到这次的抑止时间，若已超过了配置的抑止时间，则使用配置的延时时间来启动该计时器。若尚未超过配置的抑止时间则使用配置的抑止时间来计算所需要的延时时间。若该延时时间小于配置的延时时间则使用配置的延时时间，否则直接使用计算出的延时时间启动** SPF **计算。**

模式：OSPF **配置模式**

命令：timers spf <delay> <hold>

配置 SPF **计算间隔的** delay **及** hold **值**

命令：no timers spf

恢复为缺省值

参数：delay **表示计算** SPF **时需要延时的时间；**hold **表示两次** SPF **计算之间需要抑止的时间。**

缺省：delay **为** 5s **；**hold **为** 10 **秒**

11.2.8 配置管理距离

路由器上可同时运行多个路由协议，如何在多个路由协议学习的路由信息中选择，就需要使用管理距离。当不同的协议发现同一条路由时，管理距离小的，优先被选择。

模式：OSPF **配置模式**

命令：distance <distance>

配置管理距离

命令：no distance <distance>

恢复管理距离为缺省值

命令：distance ospf {intra-area <distance> | inter-area <distance> | external <distance>}

配置不同类型的管理距离

命令：no distance ospf

恢复三种类型的管理距离为缺省值

参数：distance **均取值** 1~255 **之间**；intra-area **表示域内路由的管理距离**；inter-area **表示域间路由的管理距离**；external **表示外部路由的管理距离。**

缺省：OSPF **协议的管理距离为** 110；**域内路由、域间路由、外部路由的管理距离为** 0。

11.2.9 引入外部路由

路由器上可运行多个动态路由协议，不同路由协议之间可共享路由信息。OSPF **把其他路由协议学习的路由看作自治系统外部的路由，由自治系统边界路由器** ASBR **引入。引入外部路由时可指定权值、权值类型等属性。**

OSPF **的路由分四种类型，其一是域内路由，其二是域间路由，这两类路由均是自治系统内的；其三是** type-1 **的外部路由，其四是** type-2 **的外部路由，这两类路由描述的是到自治系统外的目的地的路由。**type-1 **的路由是来自其他** IGP **的路由，**OSPF **认为其可信度比较高，且与自治系统内的路由权值具有可比性，所以这类外部路由的花费是路由器本身到** ASBR **的花费与** ASBR **到目的地的花费之和。**type-2 **的路由是来自其他** EGP **的路由，**OSPF **认为其可信度不太高，且其花费远大于自治系统内的路由花费，不具可比性，所以这类外部路由的花费仅使用** ASBR **到目的地的花费，而忽略路由器本身到** ASBR **的花费。**

模式：OSPF **配置模式**

命令：redistribute {kernel | connected | static | rip | isis | bgp} [metric <metric> | metric-type <type> | route-map <route-map-name> | tag <tag>]

命令：no redistribute {kernel | connected | static | rip | isis | bgp} [metric <metric> | metric-type <type> | route-map <route-map-name> | tag <tag>]

参数：第一个必选参数为可引入的外部路由的类型，有直连、静态、RIP、IS-IS、BGP；**第二个参数为引入外部路由时设置的权值，取值** 0~(2^{24}-1)**之间；第三个参数为引入的两类外部路由，分** type-1 **和** type-2，type-1 **为** IGP **路由，**type-2 **为** EGP **路由；第三个参数为引用的** route-map **的名，**route-map **在全局配置模式下配置，可参看命令手册；第四个参数为** tag，**取值** 0~(2^{32}-1)**之间，为外部路由属性。**

缺省：不引入任何外部路由协议

11.2.10 配置接口的网络类型

OSPF 协议是以本身路由器为视角的，每台路由器均将自己邻接的网络拓扑描述出来，传递给其他路由器。OSPF 根据链路层协议类型将接口链路的网络类型分为四种：一是广播类型（链路层协议是 Ethernet、FDDI etc）；二是 NBMA 非广播多路访问类型（链路层协议是 FR、ATM、HDLC、X.25 etc）；三是点到多点类型，没有一种链路层协议会被缺省认为是点到多点类型。点到多点类型必须是由其他的网络类型强制配置的。最常见的做法是将非全连通的 NBMA 改为点到多点网络。四是点到点类型（链路层协议是 PPP、LAPB、POS）。

在没有多路访问能力的广播网络上，可将接口配置成 NBMA 类型。在 NBMA 网络中并非所有路由器之间都直接可达时，可将接口配置成点到多点类型。

OSPF 协议中约定的 NBMA 网络是全连通的，非广播的，多点可达的。点到多点网络不一定是全连通的。NBMA 需要进行 DR 选择，点到多点网络中没有 DR。NBMA 网络通过指定邻居单播报文，点到多点网络多播报文。

模式：接口配置模式

命令：ip ospf network <type>

配置接口链路的网络类型

命令：no ip ospf network

恢复接口链路的网络类型为缺省值

参数：type **可选择** broadcast、non-broadcast、point-to-point、point-to-multipoint [non-broadcast]；第一种类型是广播网络，第二种类型为非广播网络，也即 NBMA，第三种类型是点到点网络，第四种类型是点到多点网络；点到多点网络又分为广播型和非广播型网络，非广播型邻居不能自动发现，必须指定邻居。

缺省：为广播网络

11.2.11 配置 hello 报文发送时间间隔

Hello 报文用于周期性发至邻居路由器，发现与维持邻居关系，选举 DR 及 BDR。Hello 报文的间隔可以手工配置，但需注意保持网络中邻居间的 hello 计时器间隔一致。Hello 计

时器的值与路由器收敛速度、网络负荷成反比。

模式：接口配置模式

命令：ip ospf hello-interval <seconds>

配置 hello 计时器的间隔

命令：no ip ospf hello-interval

恢复 hello 计时器间隔为缺省值

参数：seconds 取值 1~(2^{16}-1)之间，表示两次 hello 报文发送之间的时间间隔。

缺省：在广播网络和点到点网络上 hello 间隔 10 秒；在 NBMA 网络和点到多点网络上 hello 间隔 30 秒。

11.2.12 配置邻居路由器失效时间

模式：接口配置模式

命令：ip ospf dead-interval <seconds>

配置邻居失效时间

命令：no ip ospf dead-interval

恢复邻居失效时间为缺省值

参数：seconds 取值 1~(2^{16}-1)之间，表示经过 seconds 时间未接收到邻居的 hello 报文则认为该邻居已失效；每次接收到 hello 报文时均会更新邻居的 dead 计时器。

缺省：广播网络和点到点网络上邻居失效时间为 40 秒；在 NBMA 网络和点到多点网络上邻居失效时间为 120 秒；当修改了网络类型后，hello 间隔和 dead 间隔将使用缺省值。

11.2.13 配置重传时间

OSPF 是可靠的链路状态协议，表现在其交互的 LSU 报文均需要对端的应答 LSU-ack。当接收到确认报文时，方认为链路状态更新被接收。若在重传间隔内没有收到确认报文就会

向邻居重传这条 LSA。**重传间隔可手工配置，需大于一个报文在两台路由器之间传送一个来回的时间，若设置的太小，会引起不必要的重传。**

模式：接口配置模式

命令：ip ospf retransmit-interval <seconds>

配置接口的重传间隔

命令：no ip ospf retransmit-interval

恢复重传间隔为缺省值

参数：seconds **取值** 1~(2^{16}-1)**之间，表示当对端未接收到** LSA **时需要重传的间隔。**

缺省：重传间隔 5s

11.2.14 配置接口延时

在链路状态更新报文 LSU **中每条链路状态广播** LSA **均有** age **时间域，在传送前需要增加发送接口的传输时延。该参数主要考虑接口发送报文需要的时间，尤其在低速网络上需要考虑配置该参数。**

模式：接口配置模式

命令：ip ospf transmit-delay <seconds>

设置接口的传输延时

命令：no ip ospf transmit-delay

恢复接口的传输延时为缺省值

参数：seconds **取值** 1~(2^{16}-1)**之间，表示在该接口发送的** LSA **的** age **域需要增加这个时延值。**

缺省：接口的传输时延 1s

11.2.15 配置接口在 DR 选举中的优先级

在广播网络中为避免重复的点到点之间传送链路信息，需要选举出指定路由器 DR 及 BDR 负责广播网段内的链路信息。接口的优先级表示其在选举 DR 时所具有的资格，当选举发生冲突时，优先级高的首先考虑。优先级为 0 不参与选举，优先级大于 0 均是候选人，每台路由器在各自的 hello 报文中包含自己的优先级信息及自己认为的 DR，在广播网络中广播，最后选择优先级大的成为 DR。若优先级相等则比较 router ID 大者优先。

当 DR 失效后，网络中路由器又需要经历一个重新选举 DR 的过程，这需要一个时间，且在这段时间内会引起路由计算错误。BDR 的概念就是为了平滑得过渡到新的 DR。BDR 是 DR 的备份，在 DR 选举中同时选出，它也和网络中其他路由器建立邻接关系，只是网络中信息的收集与发布结点在 DR 而非 BDR，BDR 仅维护邻接的同步。当 DR 失效后，BDR 会立即成为 DR，负责收集网段内信息，而此时会启动新的过程选举 BDR，但 BDR 的选举不影响路由的计算。

模式：接口配置模式

命令：ip ospf priority <prio>

配置接口在 DR 选举中的优先级

命令：no ip ospf priority

恢复接口优先级为缺省值

参数：prio 取值 0~255，表示 DR 选择中的优先级，当为 0 时表示不参与选举。

缺省：优先级为 1

11.2.16 配置接口上发送报文的代价

网络中通过配置不同链路不同的代价来控制流量，接口的代价表示从该接口发送报文的花费。若不手工配置则 OSPF 会根据接口波特率自动计算接口代价。

模式：接口配置模式

命令：ip ospf cost <cost>

命令：no ip ospf cost

参数：cost 取值 $1\sim(2^{16}-1)$之间，表示该接口上发送报文的代价值。

缺省：接口代价 10

11.2.17 配置接口发送 DD 报文是否填 MTU 域

模式：接口配置模式

命令：ip ospf mtu-ignore

设置不检查 DD **报文中** mtu **值**

命令：no ip ospf mtu-ignore

取消不检查 DD **报文中** mtu **值**

缺省：检查 DD **报文中** mtu **值**

11.2.18 配置接口报文认证

OSPF**协议在接口上的报文认证支持明文方式和**MD5**方式。**

模式：接口配置模式

命令：ip ospf authentication <mode>

配置认证模式

命令：no ip ospf authentication

取消认证

参数：无参数表示明文认证；message-digest**表示**MD5**认证；**null**表示无认证**

命令：ip ospf authentication-key <password>

配置明文认证密码串

命令：no ip ospf authentication-key

取消明文认证密码串

参数：password **表示明文认证的密码字符串**

命令：ip ospf message-digest-key <key-id> md5 <password>

配置 MD5 **认证密码**

命令：no ip ospf message-digest-key <key-id>

取消 MD5 认证密码

参数：key-id 取值 1~255 之间，用于在密钥链中排序；password 表示密码字符串。

缺省：未配置任何认证

11.2.19 配置区域虚链路

OSPF 协议采用分层思想，将自治系统内的路由器划分成不同的组，这些组称之为区域，所有的区域并非平等并列，而是具有层次关系，其中 0.0.0.0 区域最特殊，为骨干区域，其他非骨干区域必须通过骨干区域来交换域间路由。所以所有非骨干区域必须与骨干区域连通，也即 ABR 上至少有一个接口在区域 0 中。如果因为网络拓扑的限制，某些区域无法与骨干区域保证物理上的通路，那么需要配置虚链路来保证逻辑上的通路。虚链路的两端都是 ABR，中间通过一个非骨干区域，被称为传输区域 transit area。配置虚链路时需指定传输区域的 ID 及对端 ABR 的 ID，且必须在两端的 ABR 上均配置方能生效。

当传输区域的路由被计算出来后虚链路被激活，则其在逻辑上相当于两个端点之间形成了一个点到点的连接，因此可以在其物理接口上配置接口的参数及启动认证功能。

ABR 之间传送的是单播报文，传输区域内转发该单播报文的路由器将其视作普通的 IP 报文来转发，因此可仅仅理解为传输区域内提供了一条逻辑链路，两台 ABR 之间可以交换协议报文。

模式：OSPF 配置模式

命令：area <area-id> virtual-link <router-id>

配置虚链路的传输区域及对端ID

[authentication <mode> |

配置虚链路的认证模式

authentication-key <password> |

配置虚链路明文认证密码

message-digest-key <key-id> md5 <password> |

配置虚链路MD5认证密码

hello-interval <seconds> |

配置虚链路的hello间隔

dead-interval <seconds> |

配置虚链路邻居失效时间

retransmit-interval <seconds> |

配置虚链路重传间隔

transmit-delay <seconds> |

配置虚链路接口时延

命令：no area <area-id> virtual-link <router-id>

[authentication <mode> |

authentication-key <password> |

message-digest-key <key-id> md5 <password> |

hello-interval <seconds> |

dead-interval <seconds> |

retransmit-interval <seconds> |

transmit-delay <seconds>]

取消虚链路设置

参数：area-id **表示传输区域的** ID**，可以使用点分十进制格式** A.B.C.D**，也可以使用整型数格式，取值** 0~(2^{32}-1)**之间。**router-id **表示虚链路对端路由器的** ID**，使用** A.B.C.D **格式。认证及发送接口属性均为可选项，可参考相关命令说明。**

缺省：不配置虚链路

11.2.20 配置区域路由聚合

模式：OSPF **配置模式**

命令：area <area-id> range <ip-prefix> [advertise | not-advertise]

配置聚合范围

命令：no area <area-id> range <ip-prefix> [advertise | not-advertise]

取消聚合

参数：area-id **表示区域** ID**，指定聚合该区域内路由，可以使用点分十进制格式** A.B.C.D**，也可以使用整型数格式，取值** 0~(2^{32}-1)**之间。**Ip-prefix **使用前缀格式** A.B.C.D/M **表示聚合范围。可选参数** advertise **和** not-advertise **表示是否广播聚合范围，也即** ip-prefix**。原有的网络路由均会广播。**

11.2.21 配置区域报文认证

一个区域中所有路由器的认证类型需保持一致。一个网段中所有路由器的认证密码串需保持一致。配置区域认证仅启动认证功能（明文或者 MD5)**，密码使用接口的相应配置值。可参考接口报文认证配置。**

模式：OSPF **配置模式**

命令：area <area-id> authentication [message-digest]

配置区域认证模式

命令：no area <area-id> authentication

取消区域认证

参数：area-id **表示区域** ID**，指定需认证的区域；可以使用点分十进制格式** A.B.C.D**，也可以使用整型数格式，取值** 0~(2^{32}-1)**之间。可选参数无表示明文认证，带** message-digest **表示** MD5 **认证。**

缺省：不启动区域认证

11.2.22 配置 stub 区域

模式：OSPF **配置模式**

命令：area <area-id> stub [no-summary]

配置路由器在 stub **区域内**

命令：no area <area-id> stub [no-summary]

取消路由器在 stub **区域的属性**

命令：area <area-id> default-cost <cost>

配置连接在 stub **区域的** ABR **广播路由的缺省代价**

命令：no area <area-id> default-cost

恢复缺省代价为默认值

参数：area-id **表示区域** ID**，指明哪个区域属性为** stub**；可以使用点分十进制格式** A.B.C.D**，也可以使用整型数格式，取值** 0~(2^{32}-1)**之间。**no-summary **表示不将域间路由注入** stub **区域。**

第一组命令是配置位于 stub **区域内的路由器，第二组命令是配置有连接到** stub **区域的**

接口的 ABR。

缺省：不配置 stub 区域

11.2.23 配置 nssa 区域

模式：OSPF 配置模式

命令：area <area-id> nssa [options]

配置 nssa 属性

命令：no area <area-id> nssa [options]

取消 nssa 区域属性

参数：area-id 表示区域 ID，options 详见命令手册。

缺省：不配置 nssa 区域

11.2.24 配置外部路由聚合

从其他协议引入的路由是一条一条放在 type-5 的 LSU 中广播的，使用聚合命令指定一个前缀范围，在这个范围内覆盖的路由均被抑止，只广播聚合后的这条路由。当外部路由数量巨大时，能有效的减少 LSDB 的规模。

模式：OSPF 配置模式

命令：summary-address <ip-prefix> [not-advertise | tag <tag>]

配置聚合范围及属性

命令：no summary-address <ip-prefix> [not-advertise | tag <tag>]

取消外部路由的聚合

参数：ip-prefix 使用地址前缀格式 A.B.C.D/M 表示需要聚合的路由范围；not-advertise 表示聚合后的路由不被广播；tag 为设置的 tag 值，取值 0~(2^{32}-1)之间，缺省为 0。

缺省：不聚合外部引入的路由

11.2.25 配置外部路由的缺省权值

引入外部路由时，若 redistribute **命令不指定** metric **值，使用缺省权值。**

模式：OSPF **配置模式**

命令：default-metric <metric>

配置引入外部路由时的缺省权值

命令：no default-metric [metric]

恢复引入外部路由时缺省权值为默认值

参数：metric **取值** 0~(2^{24}-1)**之间**

缺省：缺省权值为 1

11.2.26 显示信息

模式：普通模式或特权模式

命令：show ip protocols

命令：show ip protocds ospf

显示OSPF **协议信息**

命令：show ip ospf [process-id]

显示OSPF **进程信息**

参数：instance-id**为进程号，**

取值0-(2^{16}-1)**之间。**

命令：show ip ospf border-routers

显示ABR**信息**

命令：show ip ospf database <type>

显示LSDB **信息**

参数：type**为各类型**LSA**及汇总信息，详见命令手册。**

命令：show ip ospf interface [if-name]

显示OSPF **接口信息**

参数：if-name**为约定的三层接口名**

命令：show ip ospf route [count]

显示OSPF **路由表**

参数：count **表示显示路由表总条目数**

命令：show ip ospf virtual-links

显示OSPF **虚连接信息**

命令：show ip ospf neighbor [options]

显示OSPF **邻居信息**

参数：options**详见命令手册**

模式：特权模式

命令：show running-config

显示交换机当前配置，包括OSPF **配置。**

命令：show running-config ospf

显示OSPF **协议的当前配置。**

11.3 OSPF 配置示例

(1) 配置

三台交换机两两相连，分别有6 **个网段，都启用**OSPF **协议，实现三台**PC **机之间能够两两互通。要求接口在同一区域**area 0。

在交换机1**上：**

```
Switch#config t
Switch(config)#router ospf 100
Switch(config-ospf-100)#network 10.1.1.0/24 area 0
Switch(config-ospf-100)#network 10.1.2.0/24 area 0
Switch(config-ospf-100)#network 192.168.1.0/24 area 0
```

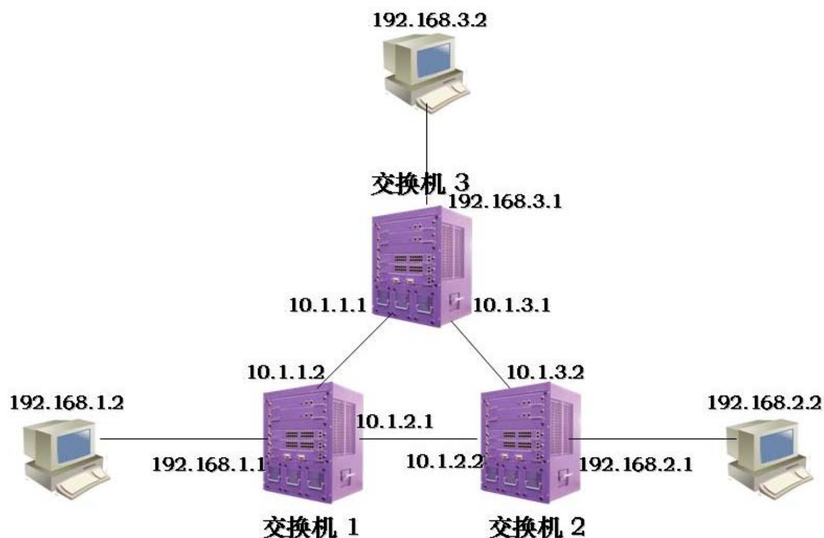
在交换机2**上：**

```
Switch#config t
Switch(config)#router ospf 100
Switch(config-ospf-100)#network 10.1.2.0/24 area 0
Switch(config-ospf-100)#network 10.1.3.0/24 area 0
Switch(config-ospf-100)#network 192.168.2.0/24 area 0
```

在交换机3**上：**

```
Switch#config t
Switch(config)#router ospf 100
Switch(config-ospf-100)#network 10.1.1.0/24 area 0
```

```
Switch(config-ospf-100)#network 10.1.3.0/24 area 0
Switch(config-ospf-100)#network 192.168.3.0/24 area 0
```



(2) 验证

```
show ip ospf database
show ip ospf interface
show ip ospf neighbor
show ip route ospf
show ip ospf route
```

第12章 配置VRRP

本章主要包括以下内容：

- VRRP **介绍**
- VRRP **配置**
- VRRP **配置示例**

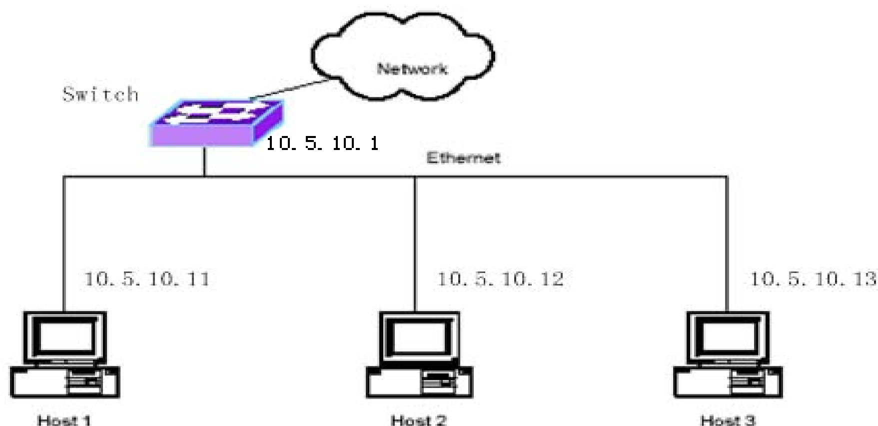
12.1 VRRP 介绍

VRRP 是虚拟路由器冗余协议的简称，是一个重要的三层可靠性协议，用于缺省网关的冗余备份。本节对 VRRP 协议进行一个详细的描述，主要包括以下内容：

- VRRP 概述
- VRRP 术语
- VRRP 协议交互
- 虚拟主路由器的选举
- 虚拟路由器的状态

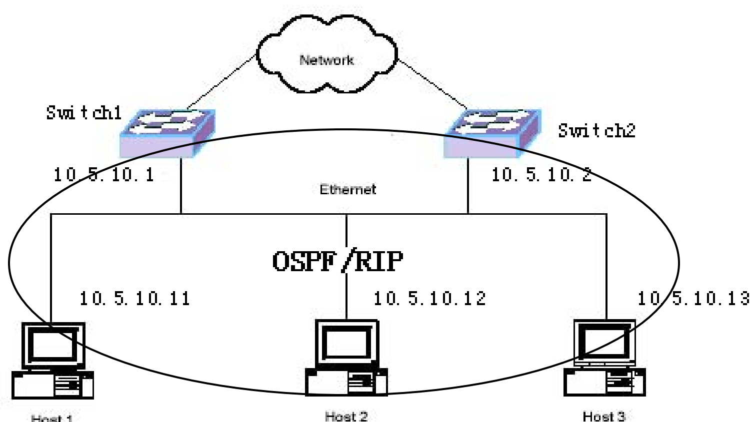
12.1.1 VRRP 概述

下图是一个典型的内部网组网方案。交换机的一个接口与外部网络相连，一个接口与内部网络相连，与内部网相连的接口的 IP 地址是 10.5.10.1，主机 1，2，3 都配置了 IP 地址，都在网段 10.5.10.0/24 内。主机 1，2，3 上都配置了一个默认网关，下一跳指向交换机，下一跳的 IP 地址是 10.5.10.1。这样，主机发送一个目的 IP 地址不在本网段内的报文会匹配缺省路由而发送到交换机，交换机再把报文转发出去，交 换机也把外部网络发来的报文转发给相应的主机，这样主机就实现了与外部网络的通信。



在上面这种组网方案中，主机与外部网络之间的通信只能通过这个唯一的交换机，当交换机出现故障时，所有的主机都与外部中断。为了解决这个问题，有一种解决方案就是把一台交换机扩展为两台或多台交换机，在主机和交换机之间都运行动态路由协议 OSPF

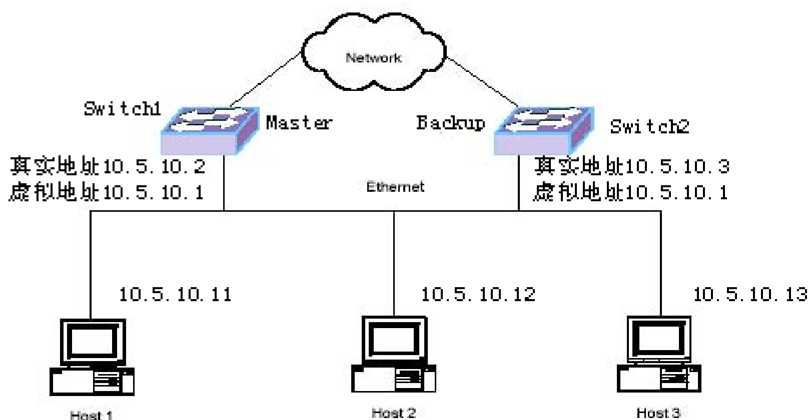
或 RIP，如下图。



当主机运行了动态路由协议后，主机上能够学习到外部网络所有的路由，主机在与外部网络的通信时，根据报文的目的 IP 地址查找路由得到下一跳来决定报文是发送给 switch1 还是 switch2。当其中的一台交换机出现故障时，主机中的路由在很短的时间内能够重新学习，路由的下一跳会指向没有故障的路由器，这样，主机与外部网络的通信不会中断。

但是，在主机上实现动态路由协议是不现实的。对于主机来说，运行动态路由协议负载太大，对于网络来说，主机上运行动态路由协议会造成网络上过多的不必要的数据流量，况且有些主机根本就不支持动态路由协议。

为了根本解决这个单点故障的问题，VRRP 协议是最好的选择。VRRP 协议是专门针对这个问题而提出来的。如下图所示，Switch1 和 Switch2 组成一个虚拟路由器，两个交换机的接口的真实 IP 地址是不一样的，但是有一个共同的虚拟 IP 地址 10.5.10.1，主机的默认网关设置为虚拟 IP 地址 10.5.10.1。当 Switch1 是虚拟主交换机时，主机与外部网络的通信是通过 Switch1 来转发，但当 Switch1 出现故障时，Switch2 接替 Switch1 成为虚拟主交换机，主机与外部网络的通信通过 Switch2 来转发。使用 VRRP 协议，主机只需要设置默认网关，而不需要在主机上运行别的协议，主机的负载小，而网络上只需要增加很少的 VRRP 协议流。



12.1.2 VRRP 术语

下面介绍几个经常要用到的术语：

1）VRRP

Virtual Router Redundancy Protocol 的缩写，虚拟路由器冗余协议，是一种缺省网关的容错协议，可提高网络的可靠性。

2）Virtual Router

虚拟路由器，一个抽象对象，基于子网接口，包括一个虚拟路由器标识符（VRID）和一个或多个 IP 地址，这个（些）IP 地址又称为虚拟 IP 地址，虚拟 IP 地址作为主机的默认网关。

3）VRRP Router

VRRP 路由器，即运行 VRRP 协议的路由器，一个 VRRP 路由器可以加入到一个或多个虚拟路由器中。

4）IP Address Owner

IP 地址拥有者，虚拟路由器的虚拟 IP 地址与接口的真实 IP 地址相同的 VRRP 路由器。

5）Virtual Router Master

虚拟主路由器,负责转发通过虚拟路由器的三层数据包,对虚拟路由器的IP 地址的 ARP 请求进行回应。如果某个 VRRP 路由器是 IP 地址拥有者，则它总是虚拟主路由器。

6）Virtual Router Backup

虚拟备份路由器，不转发三层数据包，不应答虚拟 IP 地址的 ARP 请求，当虚拟主路由器出现故障时接替虚拟主路由器的工作。

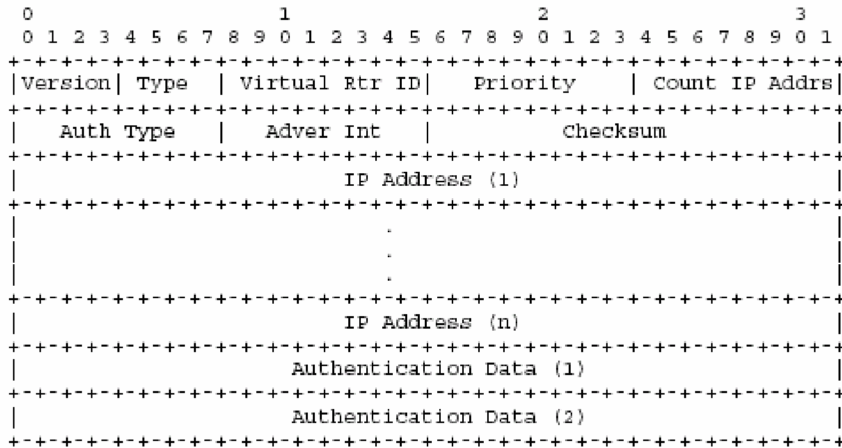
为了更好地理解这几个术语，要注意以下几点：

- 一个交换机可包括多个接口，可在多个接口子网上启动 VRRP 协议。
- 一个接口子网上可存在一个或多个虚拟路由器。
- 一个 VRID 标识一个虚拟路由器，在一个接口子网上不同的虚拟路由器的 VRID 不同。

- **在一个交换机上的不同接口子网上的虚拟路由器的** VRID **可以相同。**

12.1.3 VRRP 协议交互

VRRP **协议包封装在** IP **包内，**VRRP **报文头如下图：**



1）VRRP **包的** MAC **帧头字段**

源 MAC **地址： 虚拟路由器的虚拟** MAC **地址，为** 00-00-5e-00-01-{VRID}，VRID **是虚拟路由器标识符。例如虚拟路由器的** VRID **为** 1，**则虚拟** MAC **地址为** 00-00-5e-00-01-01。

目的 MAC **地址：**VRRP **组播** MAC **地址，为** 01-00-5e-00-00-12。

2）VRRP **包的** IP **包头字段**

源 IP **地址：发送** VRRP **包的接口的主** IP **地址。**

目的 IP **地址：组播** IP **地址** 224.0.0.18，**不能够做三层转发。**

TTL：255，**为了防止远端** VRRP **包攻击。**

Protocol：112。

3）VRRP **包头字段**

Version：2。

Type：VRRP **包的类型，只支持一种类型：**1 --- ADVERTISEMENT，VRRP **通告包。**

VRID：标识一个虚拟路由器。

Priority：对于此虚拟路由器来说，发送的 VRRP 路由器的优先级。

Count IP Addrs：虚拟 IP 地址的个数，一个虚拟路由器中可以有多个虚拟 IP 地址。

Auth Type：一个虚拟路由器中的 VRRP 路由器之间的认证方法。

Advertisement Interval：通告的间隔时间，缺省为 1 秒。

Checksum：校验和，从 VRRP 包头的 Version 算起。

IP Address(es)：一个或多个虚拟 IP 地址。

Authentication Data：认证的数据。

4）VRRP 优先级

一个虚拟路由器中的每一个 VRRP 路由器都需要配置一个优先级 priority。优先级的范围从 0 到 255，其中 0 和 255 有特殊的用途，可配置的优先级范围从 1 到 254，缺省为 100。优先级的值越大，优先级越高，越有可能成为虚拟主路由器。

在一个虚拟路由器中当某个 VRRP 路由器是 IP 地址拥有者时，它的优先级是 255。

当虚拟主路由器需要通告给其它备份路由器它不再是主时，发送优先级为 0 的 VRRP 包给其它备份路由器，这样可以快速触发其它备份路由器成为虚拟主路由器。

5）VRRP 认证

VRRP 协议提供了三种认证方法，在实际使用时可以根据网络的安全性要求来选择不同的认证方法。

0 --- No Authentication

不做认证

1 --- Simple Text Password

简单口令认证

2 --- IP Authentication Header

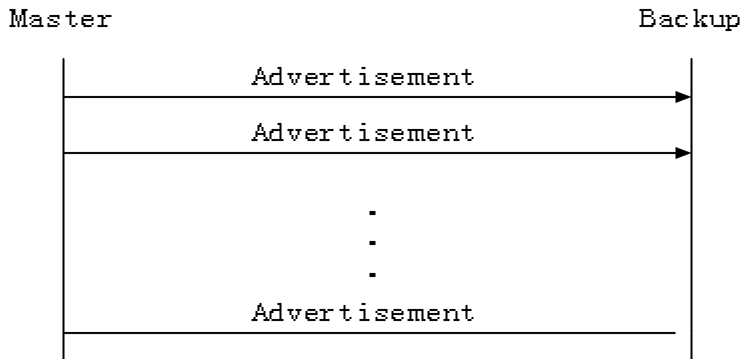
IP 认证头，通过 HMAC-MD5 方法计算消息摘要

在安全性不高的网络可以采用不做认证或简单口令认证方法，在安全性高的网络采用 HMAC 认证方法。

对于 0 和 2 认证方法，Authentication Data 字段填 0，对于 1 认证方法，Authentication Data 字段填口令。对于 2 认证方法，消息摘要填在 IP Authentication Header 字段，也就是说在 IP 头加上 AH 字段。

6）VRRP **包交互**

VRRP **协议只有一种类型的包，**ADVERTISEMENT **通告包。在一个虚拟路由器中，虚拟主路由器每隔** Advertisement Interval **时间（缺省为** 1 **秒）发送一个通告包。虚拟备份路由器根据收到的** VRRP **通告包来决定是否需要状态迁移。主与备的协议交互如下图：**



12.1.4 虚拟主路由器的选举

在一个虚拟路由器中虚拟主路由器的选择由以下因素来决定：

- IP **地址所有者**

如果一个 VRRP **路由器是** IP **地址所有者（它的接口** IP **地址与虚拟** IP **地址相同），如果该路由器工作正常，它就是虚拟主路由器。**

- VRRP **优先级**

工作正常的优先级最高的 VRRP **路由器成为虚拟主路由器。可配的优先级范围从** 1 **到** 254**，**IP **地址所有者的路由器的优先级为** 255**，当虚拟主路由器通知虚拟备份路由器自己不再是主时，在** VRRP **包中给定优先级** 0。

- **接口的实际** IP **地址大小，当优先级相同时，接口的实际** IP **地址大的** VRRP **路由器成为虚拟主路由器。**

在以下情况下，虚拟路由器中会出现主备切换：

1）**当虚拟主路由器出现故障时，会出现主备切换，这种情况下又有两种可能性：**

- **如果虚拟主路由器还能够活动，会发送一个优先级为** 0 **的** VRRP **包，备份路由器收**

到这个包后在 Skew_Time **时间内没有收到虚拟主路由器的** VRRP **包后会切换为虚拟主路由器。这种情况切换速度比较快，在** 1 **秒以内能够实现切换。**

- **如果虚拟主路由器不能活动，虚拟备份路由器在** Master Down Interval **时间内没有收到虚拟主路由器的** VRRP **包后会切换为虚拟主路由器。**

Master_Down_Interval = (3 * Advertisement_Interval) + Skew_Time

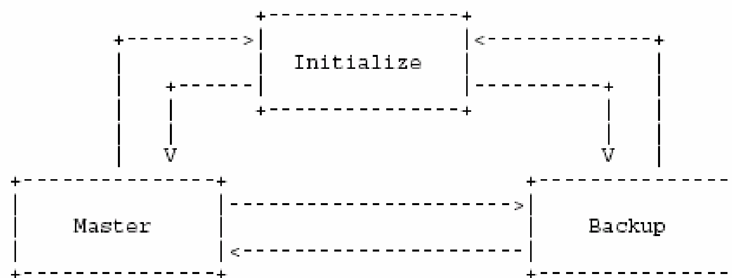
Skew_Time = ((256 – Priority) / 256)

2）**当虚拟主路由器不是** IP **地址拥有者，而现在有一个** IP **地址拥有者的路由器加入网络，此时该路由器会成为虚拟主路由器，出现主备切换。**

3）**当一个** VRRP **路由器加入网络，如果该路由器的优先级比虚拟主路由器还高并且是抢占模式（配置变量** Preempt_Mode **为** TRUE**）时，该路由器会成为虚拟主路由器，出现主备切换。**

12.1.5 虚拟路由器的状态

在一个虚拟路由器中的每一个 VRRP **路由器都执行一个状态机。状态机的迁移如下图：**



1）Initialize **状态**

Initialize **状态是一个虚拟路由器的初始状态，在这种状态下等待** Startup **启动事件。如果在这种状态下收到** Startup **事件时处理如下：**

- **如果此** VRRP **路由器的优先级是** 255**（也就是** IP **地址拥有者），该路由器成为虚拟主路由器，迁移到** Master **状态。**
- **否则，该路由器成为虚拟备份路由器，迁移到** Backup **状态。**

路由器迁移到 Master **状态后做的动作如下：**

- **发送一个** VRRP **通告包。**

- 广播一个 ARP 请求，包含虚拟 IP 地址和对应的虚拟 MAC 地址。
- 设置 Adver_Timer 定时器，定时间隔为 Advertisement_Interval。

路由器迁移到 Backup 状态后做的动作如下：

- 设置 Master_Down_Timer 定时器，定时间隔为 Master_Down_Interval。

2）Backup 状态

Backup 状态的目的是监控虚拟主路由器的可用性和状态，随时接替虚拟主路由器的工作。

如果收到一个 Shutdown 事件，取消 Master_Down_Timer 定时器，回到 Initialize 状态。

如果 Master_Down_Timer 到期，成为虚拟主路由器，迁移到 Master 状态。

如果收到一个 VRRP 通告包，存在以下几种情况：

- 如果 VRRP 包中的优先级字段为 0，设置 Master_Down_Timer，定时间隔为 Skew_Time。
- 否则，如果抢占模式（Preempt_Mode）为 FALSE 或者 VRRP 包中的优先级 >= VRRP 路由器的优先级，重设 Master_Down_Timer，定时间隔为 Master_Down_Interval。
- 否则，丢弃 VRRP 包。

3）Master 状态

处于 Master 状态的 VRRP 路由器负责转发通过虚拟路由器的三层数据包。

如果收到 Shutdown 事件，取消 Adver_Timer 定时器，发送一个优先级为 0 的 VRRP 通告包，迁移到 Initialize 状态。

如果 Adver_Timer 定时器到期，发送一个 VRRP 通告包，重设 Adver_Timer 定时器。

如果收到一个 VRRP 通告包，存在以下几种情况：

- 如果包中优先级为 0，发送一个 VRRP 通告包，重设 Adver_Timer。
- 否则如果包中的优先级大于 VRRP 路由器的优先级或优先级相同，但发送该包的 IP 地址大于 VRRP 路由器的接口主 IP 地址时，取消 Adver_Timer 定时器，设置 Master_Down_Timer，迁移到 Backup 状态。
- 否则，丢弃该 VRRP 通告包。

12.2 VRRP 配置

VRRP 配置包括以下内容：

- 创建和删除虚拟路由器
- 配置虚拟路由器的虚拟 IP 地址
- 配置虚拟路由器的参数
- 启动和关闭虚拟路由器
- 查看 VRRP 信息

12.2.1 创建和删除虚拟路由器

虚拟路由器是建立在子网接口上的，并且需要指定一个 VRID。在同一个接口下，不能有两个相同 VRID 的虚拟路由器存在，而不同的接口下可以存在两个相同的 VRID 的虚拟路由器。理论上一个接口下最多可以创建 255 个虚拟路由器，而目前交换机只实现了一个接口下最多创建 4 个虚拟路由器。系统缺省情况下没有创建虚拟路由器。

当一个虚拟路由器不再需要使用时，可以删除此虚拟路由器，如果虚拟路由器已经启动了，则会先关闭虚拟路由器，再把虚拟路由器删除。

创建和删除虚拟路由器的命令如下：

命令	描述	CLI模式
router vrrp <if-name> <vrid>	在一个接口下创建一个虚拟路由器，并且进入VRRP配置模式，如果该虚拟路由器已经存在，则直接进入VRRP配置模式。第一个参数是VLAN接口名，第二个参数是VRID,范围从1到255。	全局配置模式
no router vrrp <if-name> [vrid]	删除一个接口下的所有虚拟路由器或指定的一个虚拟路由器。第一个参数是VLAN接口名，第二个参数是VRID，	全局配置模式

	如果不输入第二个参数，则删除此接口下的所有虚拟路由器，如果输入第二个参数，则删除此接口下指定的虚拟路由器。	

注意：

- **在创建虚拟路由器之前，必须先保证接口已经存在并且在接口上已经配置了** IP **地址。**
- **在删除** VLAN **接口、删除** VLAN **接口上的** IP **地址或修改** VLAN **接口的** IP **地址时，该接口上的所有虚拟路由器都会被删除。**

12.2.2 配置虚拟路由器的虚拟 IP 地址

虚拟路由器上必须配置虚拟 IP **地址，理论上一个虚拟路由器可以存在一个或多个虚拟** IP **地址，但交换机在实现时一个虚拟路由器只支持一个虚拟** IP **地址。在配置时，一个虚拟路由器中的多个** VRRP **路由器必须配置相同的虚拟** IP **地址。缺省情况下交换机没有配置虚拟** IP **地址。**

配置虚拟路由器的虚拟 IP **地址的命令如下：**

命令	描述	CLI模式
vrrp ip-address <virtual-ip>	设置虚拟路由器的虚拟IP地址。	VRRP配置模式
no vrrp ip-address	删除虚拟路由器的虚拟IP地址。	VRRP配置模式

注意：

- **当需要修改虚拟路由器的虚拟** IP **地址时，必须先删除虚拟** IP **地址，再设置虚拟** IP **地址。**
- **配置虚拟路由器的虚拟** IP **地址必须在虚拟路由器已经关闭的情况下才能成功，当虚拟路由器启动时不能配置成功。**
- **设置的虚拟** IP **地址必须与接口的主** IP **地址在同一个网段，否则配置不成功。**

- **主机** PING **不通虚拟** IP **地址，当对交换机进行网管时，使用交换机的真实的** IP **地址，不要用虚拟** IP **地址。**

12.2.3 配置虚拟路由器的参数

虚拟路由器的参数包括优先级，抢占模式，通告时间间隔，认证方法和认证数据，这些参数都有缺省值，如下表：

参数	缺省值
优先级	100
抢占模式	TRUE
通告时间间隔	1 秒
认证方法	不做认证
认证数据	无

在配置时，对于虚拟路由器的多个 VRRP **路由器，通告时间间隔，认证方法和认证数据必须配置一样，而优先级和抢占模式参数可以配置一样，也可以配置不一样。**

对于优先级，分为配置优先级和运行优先级，大部分情况下，运行优先级使用的是配置优先级，但当 VRRP **路由器是** IP **地址拥有者时，运行优先级为** 255**，不使用配置优先级。**

对于认证方法，交换机目前只实现了不做认证和简单口令认证两种方式，而对于 IP **认证头方式没有实现。**

配置虚拟路由器的参数的命令如下表：

命令	描述	CL模式
vrrp priority <value>	设置虚拟路由器的优先级，优先级的范围为3到254,优先级1和2预留，有其它用途。	VRRP配置模式
vrrp preempt {true | false}	设置虚拟路由器的抢占模式，TRUE表示进行抢占，FALSE表示不进行抢占。	VRRP配置模式
vrrp advertisement-interval <interval>	设置虚拟路由器的通告时间	VRRP配置模式

	间隔，范围从1到255,单位为秒。	
vrrp authentication none	设置虚拟路由器的认证方法为不做认证。	VRRP配置模式
vrrp authentication simple-password <key>	设置虚拟路由器的认证方法为简单口令认证，并且要设置认证数据，即口令。口令不能超过8个字节。	VRRP配置模式

注意：

- 配置虚拟路由器的参数必须在虚拟路由器已经关闭的情况下才能成功，当虚拟路由器启动时不能配置成功。

12.2.4 启动和关闭虚拟路由器

当创建了虚拟路由器并且设置了虚拟 IP 地址和参数后，虚拟路由器并没有真正运行，还处于 Initialize 状态。启动虚拟路由器会启动协议的运行，给协议发送一个 Startup 事件，状态机迁移到 Master 状态或者 Backup 状态。关闭虚拟路由器会关闭协议的运行，给协议发送一个 Shutdown 事件，状态迁回到 Initializ e 状态。

在启动虚拟路由器前必须保证已经配置了虚拟 IP 地址。在虚拟路由器启动的情况下，如果需要修改虚拟 IP 地址或者参数，必须先关闭虚拟路由器再进行配置，配置完成后再启动虚拟路由器。

启动和关闭虚拟路由器的命令如下：

命令	描述	CL模式
enable vrrp	启动虚拟路由器。	VRRP 配置模式
disable vrrp	关闭虚拟路由器。	VRRP 配置模式

12.2.5 查看 VRRP 信息

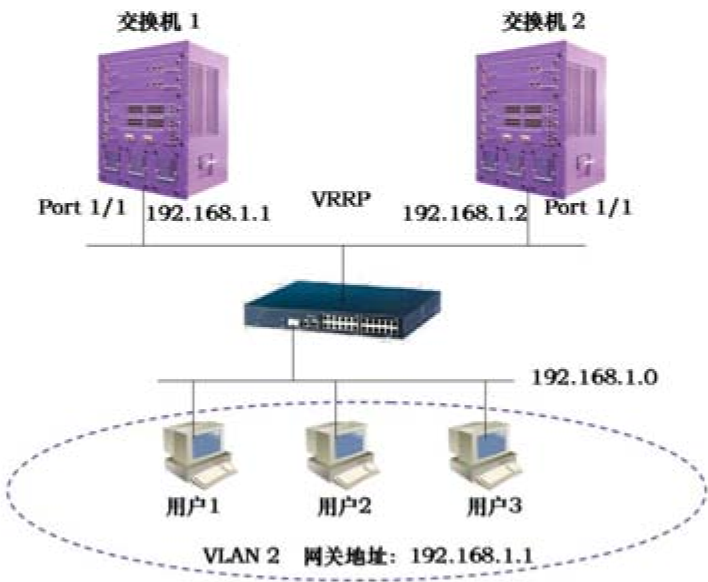
通过命令可以查看到 VRRP 的运行状态信息和配置信息，查看 VRRP 信息的命令如下：

命令	描述	CL模式
show vrrp [if-name] [vrid]	如果不输入参数，显示所有的虚拟路由器的信息，如果只输入第一个参数，显示某个接口下的虚拟路由器的信息，如果输入两个参数，则显示某个接口下的指定的虚拟路由器。	普通模式，特权模式
show running-config	查看系统的当前配置，可以查看到VRRP 的配置。	特权模式

12.3 VRRP 配置示例

(1) 配置

在两台交换机上启用VRRP功能，为局域网中的用户提供三层路由冗余功能，消除网络中的路由故障，设置交换机1为主用交换机Master，交换机2为备份交换机Backup。



交换机1**上的配置：**

```
Switch#config t
Switch(config)#vlan database
Switch(config-vlan)#vlan 2
Switch(config-vlan)#exit
Switch(config)#interface ge1/1
Switch(config-ge1/1)#switchport access vlan 2
Switch(config-ge1/1)#exit
Switch(config)#interface vlan2
Switch(config-vlan2)#ip add 192.168.1.1/24
Switch(config-vlan2)#exit
Switch(config)#router vrrp vlan2 1
Switch(config-vrrp)#vrrp ip-address 192.168.1.1
Switch(config-vrrp)#enable vrrp
```

交换机2**上的配置：**

```
Switch#config t
Switch(config)#vlan database
Switch(config-vlan)#vlan 2
Switch(config-vlan)#exit
Switch(config)#interface ge1/1
Switch(config-ge1/1)#switchport access vlan 2
Switch(config-ge1/1)#exit
Switch(config)#interface vlan2
Switch(config-vlan2)#ip add 192.168.1.2/24
Switch(config-vlan2)#exit
Switch(config)#router vrrp vlan2 1
Switch(config-vrrp)#vrrp ip-address 192.168.1.1
Switch(config-vrrp)#enable vrrp
```

(2) 验证:

通过以下命令查看VRRP**的信息：**

```
show running-config
show vrrp
show vrrp vlan2
```

第13章 配置DHCP RELAY

本章主要包括以下内容：

- DHCP RELAY **介绍**
- DHCP RELAY **配置**
- DHCP RELAY **配置示例**

13.1 DHCP RELAY 介绍

DHCP（动态主机配置协议 Dynamic Host Configuration Protocol）是 BOOTP 的增强版本，为网络上的主机动态配置网络环境，分为服务器端和客户端。服务器端集中管理 IP 网络资料并处理客户端的请求，动态配置客户端的 TCP/IP 环境。DHCP 工作时，至少有一台服务器在网络上，它可以监听网络上主机的 DHCP 请求，并协商 TCP/IP 的参数。其分配有自动和动态两种方式。自动方式，一旦客户端获得 IP 地址后就永久使用该地址。动态方式，客户端获得的 IP 地址有一租约，一旦租约到期就需要释放该 IP；也可以提前续约，或租用其他 IP。动态分配能有效解决实际 IP 不足的问题。

DHCP 的工作过程：

如果客户端是第一次登录网络，它无任何 IP 资料，会广播一 Discover 报文，源地址 0.0.0.0，目的地址 255.255.255.255。若服务器无响应，则根据一定间隔发出四次的 Discover 请求。

服务器接收到 Discover 则选择一空闲 IP 回应客户端 Offer 报文。

若网络上存在多台服务器，客户端会接收到多个 Offer 报文，一般选择最先到达的 Offer，并广播 Request 报文，告诉所有服务器它已接收哪台服务器提供的 IP 地址了。

如果客户端通过 ARP 发现该 IP 已被使用，则发 Decline 报文给服务器，拒绝该 Offer；并重新启动 Discover 过程。

服务器接收到 Request 报文会给客户端发送 Ack 报文确认该租约生效。

若客户端已经申请到 DHCP 的租约，一般无需再使用 Discover 过程了。在租约到期之前使用已经租用的 IP 向服务器发送 Request 续约，服务器会尽量让客户端使用原来的 IP，如果没问题的话，服务器回应 Ack 报文确认。若该 IP 已经被其他客户端使用，则服务器回应 Nack 报文拒绝该续约请求。

客户端可以使用 Release 报文主动解除租约。

工作站在开机时发出 Request 请求；在租约一半时会再发 Request 请求，若无确认仍可使用该 IP；在租约 3/4 时还会发 Request 请求，若这时无确认的话将不能再使用这个 IP 了。

Discover 报文是以广播方式发布的，只能在同一网段内，路由器不会将广播报文扩散出去。当服务器与客户端不在同一网段，客户端还未获得 IP 环境设定，也不知道路由器的位置，这时 Discover 报文是无法到达服务器的。为了解决这个问题，可使用 DHCP relay 的功能，让路由器来中转 DHCP 的协议报文，使得 DHCP 可跨网段运作。

13.2 DHCP RELAY 配置

DHCP relay 功能多与接口相关，实现 DHCP 跨网段的协议报文转发，在接口模式下进行相关配置。

DHCP-relay 的配置包括：

- 启动接口的 DHCP-relay 功能
- 配置接口对应的 DHCP server

13.2.1 启动接口的 DHCP-relay 功能

模式：接口配置模式

命令：dhcp relay　　　　在接口上打开 dhcp relay 协议

命令：no dhcp relay　　　关闭接口上 dhcp relay 协议

缺省：不打开 dhcp relay 协议；使用该命令启动接口收发 dhcp 协议报文。

13.2.2 配置接口对应的 DHCP server

模式：接口配置模式

命令：dhcp server <ip-address>

配置服务器 IP，从该接口接收的报文发往指定服务器。

命令：no dhcp server <ip-address>

删除指定服务器

命令：no dhcp server

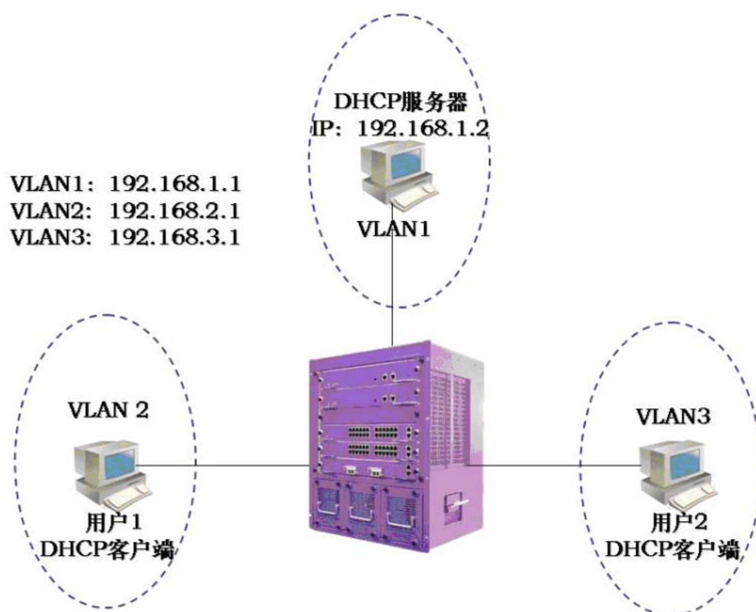
删除服务器列表

参数：<ip-address>表示服务器的 IP 地址；DHCP-relay 协议主要用于第三层的 DHCP 报文中继，该命令将指定的服务器绑定到相应接口，也即从该接口接收到的协议报文均发往指定的服务器。这样不同的网段可以分配不同的服务器。

13.3 DHCP RELAY 配置示例

(1) 配置

需要对交换机进行DHCP中继转发配置，使交换机能够路由转发用户1和用户 2的DHCP请求和DHCP服务器的DHCP回复确认信息。使用户1和用户 2能够通过在不同网段的DHCP服务器获得合法IP地址，从而接入网络。




```
Switch#config t
Switch(config)#interface vlan1
Switch(config-vlan1)#ip address 192.168.1.1/24
Switch(config-vlan1)#dhcp relay
Switch(config-vlan1)#interface vlan2
Switch(config-vlan2)#ip address 192.168.2.1/24
Switch(config-vlan2)#dhcp relay
Switch(config-vlan2)#dhcp server 192.168.1.2
Switch(config-vlan2)#interface vlan3
Switch(config-vlan3)#ip ad 192.168.3.1/24
Switch(config-vlan3)#dhcp relay
Switch(config-vlan3)#dhcp server 192.168.1.2
```

(2) 验证

show running-config

第14章 配置系统日志

本章主要包括以下内容：

- **系统日志介绍**
- **系统日志配置**

14.1 系统日志介绍

系统日志模块是交换机的一个重要组成部分，它用来记录整个系统的运行情况，异常行为及用户的操作行为，帮助管理员及时了解和监控系统的工作情况。系统日志模块管理系统的所有来源于正在运行的各个模块的日志信息，对日志信息进行收集，分类，存储和显示输出。

在日志系统中，还有一个重要的 debugging 的功能。系统日志与 debugging 配合，可以帮助管理员或其他技术人员监控网络的运行情况，调试和诊断网络中出现的故障。管理员可以方便地选择需要调试的内容，通过观察 debugging 输出的日志信息，来定位和解决设备或网络的故障。

本节主要包括以下内容：

- 日志信息的格式
- 日志的存储
- 日志的显示
- debugging 工具

14.1.1 日志信息的格式

日志信息的格式如下：

时间戳 优先级: 模块名: 日志内容

时间戳与优先级之间有一个空格，优先级与模块名之间有一个冒号和一个空格，模块名与日志内容之间有一个冒号和一个空格。

日志信息的格式的例子如下：

2006/05/20 13:56:34 Warning: MSTP: Port up notification received for port ge2/2

在这条日志信息中，时间戳是 2006/05/20 13:56:34；优先级是 Warning；模块名是 MSTP；日志内容是 Port up notification received for port ge2/2。

1）时间戳

时间戳的格式：年/月/日 小时:分:秒。

小时采用的是 24 小时制的，从 0 到 23。

时间戳记录的是这条日志信息产生的时间，使用的是交换机的系统时间。系统时间在交换机出厂时已经设置，管理员也可以修改，设备断电后系统时间依然能够运行。

2）优先级

优先级记录这条日志信息的重要程度，根据日志信息的重要程度把日志信息分为四级，优先级从高到低的顺序为：Critical，Warning，Informational 和 Debugging。优先级的描述如下表：

优先级	描述
Critical	严重的错误
Warning	一般的错误，警告，非常重要的提示
Informational	重要的提示，一般的提示，诊断信息
Debugging	调试信息

3）模块名

模块名记录这条日志信息产生的模块，下表列出了一些主要的产生日志信息的模块：

模块名	描述
CLI	命令行接口模块
MSTP	多实例生成树协议模块
VLAN	VLAN 功能模块
OSPF RIP	OSPF 协议模块 RIP 协议模块
ARP	ARP 协议模块
IP	IP 协议模块
ICMP UDP	ICMP 协议模块 UDP 协议模块
TCP	TCP 协议模块
VRRP	VRRP 协议模块
DHCP-relay	DHCP RELAY 协议模块

IGMP	IGMP 协议模块
brd_mgr	模块管理模块

4）日志内容

日志内容是一个短语或句子，代表该日志信息的内容大意，管理员通过阅读日志内容可以知道系统发生了什么事情。

14.1.2 日志的存储

日志的存储一般有三种方式，分别是：

- 日志存储在内存中。
- 日志存储到 NVM 中。
- 日志存储到服务器中。

根据日志的优先级在内存中存在四张日志表，每张表存放一种优先级的日志信息，也就是根据日志的优先级把日志分成四类，每类日志存在一个单独的日志表中。每张日志表都有 1K 个条目，可以存放 1K 条日志信息，当日志表满时后面的日志覆盖时间最久的日志信息。这种存储方式有一个问题，当系统重新启动后这些日志信息都没有了，管理员在系统崩溃的时候没法看到日志信息，没法定位问题。

对于重要的日志信息，如优先级为 Critical 和 Warning 的日志信息，可以把这些日志信息存储到系统的 NVM 中。这种存储方式在系统重启后，NVM 中的日志信息还能够被保留，便于管理员在系统崩溃时定位问题。但这种存储方式有一个问题是由于 NVM 的容量限制，在 NVM 中存储的日志信息条目非常有限。

还有一种比较好的方式是把日志信息存储到服务器中，使用 SYSLOG 协议可以实现，日志信息可以实时地发送到服务器上，服务器保存这些日志信息并显示在一个界面上。这种存储方式不仅便于用户查看日志信息，而且容量巨大，可以把大量的日志信息都存储在服务器上。

目前系统只支持把日志信息存储到内存中，不支持把日志信息存储到 NVM 或服务器中。

14.1.3 日志的显示

日志的显示有两种方式：手工显示和实时显示。手工显示就是用户通过输入命令的方式把日志信息显示出来，实时显示就是当产生日志信息时，日志信息直接输出到终端上，用户可以及时看到。

对于手工显示的方式，用户可以查看所有的日志信息，也可以查看一个优先级的日志信息。日志信息的显示顺序是最后产生的日志信息放在最前面，这样用户可以先看到交换机最近的运行状态。

对于实时显示的方式，用户必须打开终端实时显示开关。如果开关是打开的，产生的日志信息不仅写入到日志表中，而且日志信息也输出到终端上，如果开关是关闭的，则日志信息不会实时显示在终端上。系统目前只能把日志信息实时输出到 Console 终端上，不支持把日志信息输出到 Telnet 终端上。

14.1.4 debugging 工具

debugging 是用于设备和网络的诊断工具，对系统和模块的数据包收发，模块的状态机变化等进行跟踪，可以让管理员了解和监控系统和模块的运行过程，如果网络或设备出现了异常情况，可以通过 debugging 工具跟踪到。

debugging 工具提供了丰富的开关，通过控制这些开关，管理员可以跟踪自己感兴趣的内容。当设备或网络出现异常时，管理员可以打开与此异常相关的 debugging 开关，通过跟踪系统和模块的执行过程找到问题所在。

当某个 debugging 开关打开时，系统会产生相关的日志信息，这些日志信息会写到相应的日志表中。一般情况下，debugging 产生的日志信息的优先级是 Informational。当终端实时显示开关打开时，这些日志信息会实时输出到终端上。当 debugging 开关关闭时，系统不会产生相关的日志信息。

14.2 系统日志配置

系统日志配置包括以下内容：

- 配置终端实时显示开关
- 查看日志信息
- 配置 debugging 开关

- 查看 debugging 信息

14.2.1 配置终端实时显示开关

缺省情况下终端实时显示开关是关闭的，系统产生的日志信息都写入到日志表中，但不会实时显示在终端上。系统中也有些日志信息是不受此开关的限制，如模块上线和下线信息，这些日志信息总会实时输出到 Console 终端上。

终端实时显示开关是与系统日志的优先级对应的，如果某个优先级的终端实时显示开关打开，则该优先级的日志信息会实时显示在终端上，如果某个优先级的终端实时显示开关没有打开，则该优先级的日志信息不会实时显示在终端上。

交换机目前只能在 Console 终端上实时显示日志信息，不能在 Telnet 终端上实时显示日志信息。

当用户使用 write 命令把系统当前配置存储到配置文件时，终端实时显示开关的配置不会存储到系统的配置文件中，当系统重启后这些配置将丢失，需要重新配置。

配置终端实时显示开关的命令如下表：

命令	描述	CLI 模式
log display [critical | warning | informational | debugging]	打开终端实时显示开关。如果不输入参数，打开所有优先级的终端实时显示开关，如果输入其中一个参数，打开指定的优先级的终端实时显示开关。	特权模式
no log display [critical | warning | informational | debugging]	关闭终端实时显示开关。如果不输入参数，关闭所有优先级的终端实时显示开关，如果输入其中一个参数，关闭指定的优先级的终端实时显示开关。	特权模式

14.2.2 查看日志信息

查看日志信息的命令如下表：

命令	描述	CLI 模式
show log display	显示所有优先级的终端实时显示开关的配置。	普通模式，特权模式
show log [critical | warning | informational | debugging]	显示日志表中的日志信息。如果不输入参数，显示所有的日志表的日志信息，如果输入其中的一个参数，显示指定的优先级的日志表的日志信息。	普通模式，特权模式

14.2.3 配置 debugging 开关

系统提供了丰富的 debugging 开关，涉及到多个模块，这里只列出每个模块的示意的命令，关于命令的完整格式参见命令手册。

当用户使用 write 命令把系统当前配置存储到配置文件时，debugging 开关的配置不会存储到系统的配置文件中，当系统重启后这些配置将丢失，需要重新配置。

配置 debugging 开关的示意命令如下：

命令	描述	CLI 模式
debug ip …	打开系统收发 IP 包的相关的 debugging 开关。	特权模式
no debug ip …	关闭系统收发 IP 包的相关的 debugging 开关。	特权模式
debug ip icmp …	打开系统收发 ICMP 包的相关的 debugging 开关。	特权模式
no debug ip icmp …	关闭系统收发 ICMP 包的相关的 debugging 开关。	特权模式

debug ip arp …	**打开系统收发**ARP **包的相关的** debugging **开关。**	**特权模式**
no debug ip arp …	**关闭系统收发**ARP **包的相关的** debugging **开关。**	**特权模式**
debug ip udp …	**打开系统收发**UDP **包的相关的** debugging **开关。**	**特权模式**
no debug ip udp …	**关闭系统收发**UDP **包的相关的** debugging **开关。**	**特权模式**
debug ip tcp …	**打开系统收发** TCP **包的相关的** debugging **开关。**	**特权模式**
no debug ip tcp …	**关闭系统收发** TCP **包的相关的** debugging **开关。**	**特权模式**
debug ospf …	**打开** OSPF **协议诊断的相关的** debugging **开关。**	**特权模式**
no debug ospf …	**关闭** OSPF **协议诊断的相关的** debugging **开关。**	**特权模式**
debug rip …	**打开** RIP **协议诊断的相关的** debugging **开关。**	**特权模式**
no debug rip …	**关闭** RIP **协议诊断的相关的** debugging **开关。**	**特权模式**
debug vrrp …	**打开** VRRP **协议诊断的相关的** debugging **开关。**	**特权模式**
no debug vrrp …	**关闭** VRRP **协议诊断的相关的** debugging **开关。**	**特权模式**
debug mstp …	**打开** MSTP **协议诊断的相关的** debugging **开关。**	**特权模式**
no debug mstp …	**关闭** MSTP **协议诊断的相关的** debugging **开关。**	**特权模式**
debug igmp snooping …	**打开**IGMP SNOOPING**功能诊断的相关的** debugging **开关。**	**特权模式**
no debug igmp snooping …	**关闭**IGMP SNOOPING**功能诊断的相关的** debugging **开关。**	**特权模式**

debug dhcp-relay …	**打开** DHCP REALY **协议诊断的相关的** debugging **开关。**	**特权模式**
no debug dhcp-relay …	**关闭** DHCP RELAY **协议诊断的相关的** debugging **开关。**	**特权模式**
no debug all	**关闭系统所有的** debugging **开关。**	**特权模式**

14.2.4 查看 debugging 信息

查看 debugging **信息的命令如下：**

命令	**描述**	CLI **模式**
show debugging [ip | ospf | rip | vrrp | mstp | igmp snooping | dhcp-relay]	**查看** debugging **开关配置。如果没有输入参数，查看所有模块的** debugging **开关配置，如果只输入其中一个参数，则只查看一个模块的** debugging **开关配置。如果输入的参数是** ip**，则会查看** IP**，**ICMP**，**ARP**，**UDP**，**TCP **模块的** debugging **开关配置。**	**普通模式，特权模式**